



**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»  
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ  
ТЕХНІКИ**

**ЗАТВЕРДЖЕНО:**

Методичною радою  
КПІ ім. Ігоря Сікорського  
(протокол №5 від «29» лютого 2024р.)

## **Факультетський КАТАЛОГ**

**вибіркових навчальних дисциплін  
першого(бакалаврського) рівня вищої освіти  
для освітніх програм спеціальностей:  
121 Інженерія програмного забезпечення  
123 Комп'ютерна інженерія  
126 Інформаційні системи та технології**

**УХВАЛЕНО:**

Вченою радою ФІОТ  
КПІ ім. Ігоря Сікорського  
(протокол №7 від «29» січня 2024 р.)

**Київ 2024**

## ЗМІСТ

Інструкція користувачам каталогу .....	6
Анотації вибіркових дисциплін .....	9
<b>2 курс 4 семестр.....</b>	<b>9</b>
<b>Освітній компонент 1Ф-каталогу.....</b>	<b>9</b>
Лінуксopodobні операційні системи.....	9
Емпіричні методи програмної інженерії.....	10
Основи клієнтської розробки.....	11
Прикладні задачі машинного навчання .....	11
Вступ до розробки програмного забезпечення .....	12
Основи розробки програмного забезпечення на платформі Microsoft.NET .....	13
Основи розробки програмного забезпечення на платформі Java.....	14
Основи розробки програмного забезпечення мовою програмування PHP .....	15
Основи розроблення програмного забезпечення на платформі Node.js .....	16
Практикум з Linux .....	17
<b>3 курс 5 семестр.....</b>	<b>18</b>
<b>Освітній компонент 2Ф-каталогу.....</b>	<b>18</b>
Системи масового обслуговування.....	18
Нечітка логіка.....	19
Мова програмування Java.....	20
Системне програмування C і C++.....	21
Сучасні технології розробки WEB-застосувачів на платформі Microsoft.NET .....	22
Сучасні технології розробки WEB-застосувачів на платформі Java.....	23
Сучасні технології розробки WEB-застосувачів мовою програмування PHP .....	24
Розробка програмного забезпечення на платформі Node.js .....	25
Технології програмування на C/Embedded (Сертифікатна програма).....	26
<b>Освітній компонент 3Ф-каталогу.....</b>	<b>29</b>
Життєвий цикл розробки програмного забезпечення .....	29
Методи та технології штучного інтелекту.....	29
Технології та засоби розробки комп'ютерної графіки та мультимедіа .....	30
Комп'ютерна графіка та мультимедіа .....	31
Проектування програмного забезпечення засобами UML.....	32
Розробка програмного забезпечення на платформі Java.....	33
Графічне та геометричне моделювання і створення реалістичних зображень.....	34
Технології Computer Vision .....	35
<b>Освітній компонент 4Ф-каталогу.....</b>	<b>37</b>
Технології просування сайтів в пошукових системах.....	37
Основи Front-end технологій .....	37

Розроблення застосунків з використанням SpringFramework .....	38
Технологія блокчейн.....	39
Linux.....	40
Розробка мобільних застосувань під iOS .....	41
AGILE методологія розробки програмного забезпечення(Авторський курс компанії SoftServe).....	41
<b><i>Освітній компонент 5.1Ф-каталогу (для кафедри ОТ).....</i></b>	<b>43</b>
Вступ до штучного інтелекту .....	43
Технології програмування користувацьких інтерфейсів (Front-end).....	44
Технології розроблення серверного програмного забезпечення (Back-end) .....	45
Розробка системних програм .....	47
Математичні основи захисту даних та інформаційної безпеки.....	48
<b><i>3 курс 6 семестр.....</i></b>	<b>50</b>
<b><i>Освітній компонент 5Ф-каталогу.....</i></b>	<b>50</b>
Аналіз даних з використанням мови Python.....	50
Розробка мобільних застосувань під Android .....	51
Основи Back-end технологій.....	52
Мультипарадигменне програмування .....	52
Сучасні технології розробки та управління життєвим циклом ПЗ.....	53
Технології програмування на ПЛІС (FPGA)(Сертифікатна програма).....	54
Системне мислення.....	56
Методи чисельної оптимізації.....	57
<b><i>Освітній компонент 6Ф-каталогу.....</i></b>	<b>59</b>
Штучний інтелект в задачах обробки зображень .....	59
Основи візуального програмування.....	60
WEB – орієнтовані технології. Основи Frontend та Backend розробок.....	60
Робота з даними в хмарних середовищах .....	63
Сучасні операційні системи.....	64
Технології програмування на мові Python .....	64
Розподілені інформаційні системи .....	65
DevOps: стратегії розгортання програмного забезпечення.....	66
<b><i>Освітній компонент 7Ф-каталогу.....</i></b>	<b>68</b>
Обробка та аналіз текстових даних на Python.....	68
Криміналістика в програмно-орієнтованих системах .....	69
Бази даних безпілотних систем та автономної робототехніки.....	70
Сучасні технології розробки WEB-застосувань з використанням мови Python .....	71
Комп'ютерна графіка та обробка зображень .....	72
Комп'ютерна графіка та мультимедіа .....	73

Управління IT-інфраструктурними проектами (Сертифікатна програма, авторський курс компанії Global Logic).....	74
Паралельні та розподілені обчислення .....	76
<b>Освітній компонент 8Ф-каталогу.....</b>	<b>78</b>
Технології та протоколи мультисервісних мереж.....	78
Програмні засоби проектування та реалізації нейромережових систем .....	79
Професійне використання SQL та PL/SQL на прикладі РСУБД Oracle 11g.....	80
Сучасні технології розробки WEB-застосовувань на платформі .NET (2 частина) .....	81
Методи та системи штучного інтелекту .....	82
Цифрова обробка сигналів .....	83
Проектування та реалізація операційних систем .....	84
Системне програмування на Асемблері .....	85
<b>4 курс 7 семестр.....</b>	<b>87</b>
<b>Освітній компонент 8.1 Ф-каталогу(для груп ІО) .....</b>	<b>87</b>
Технології та протоколи мультисервісних мереж.....	87
Програмні засоби проектування та реалізації нейромережових систем .....	88
Професійне використання SQL та PL/SQL на прикладі РСУБД Oracle 11g.....	89
Цифрова обробка сигналів .....	90
Технології Computer Vision .....	91
<b>Освітній компонент 9Ф-каталогу.....</b>	<b>93</b>
Основи WEB – технологій .....	93
Системи управління мережами .....	93
Інженерна комп'ютерна графіка.....	95
Ігровий штучний інтелект.....	96
Створення інформаційних систем (ІС) на базі Low-code платформ.....	97
Мобільні комп'ютерні мережі.....	97
Основи комп'ютерного моделювання.....	98
Технології Data Science .....	100
<b>Освітній компонент 10Ф-каталогу.....</b>	<b>102</b>
Імітаційне моделювання .....	102
Протоколи і алгоритми електронного голосування.....	103
Технології інтернет речей .....	104
Реактивне програмування.....	105
Побудова IP-мереж на базі обладнання Juniper .....	106
Системне програмування в середовищі Unix .....	107
Основи науки про дані .....	108
Тестування та контроль якості (QA) вбудованих систем (Сертифікатна програма, авторський курс компанії Global Logic) .....	110
<b>Освітній компонент 11Ф-каталогу.....</b>	<b>113</b>

Проектування мікропроцесорних систем на базі Arduino та Raspberry PI .....	113
Інфраструктура програмного забезпечення WEB-застосувань .....	113
Сучасні технології розробки WEB-застосувань на платформі .NET (3 частина) .....	114
Системи автоматизованого проектування комп'ютерних систем .....	115
Розробка програмного забезпечення. Проектний та програмний менеджмент.....	117
Мережеве управління та протоколи.....	117
<b>4 курс 8 семестр.....</b>	<b>119</b>
<b>Освітній компонент 12Ф-каталогу.....</b>	<b>119</b>
Алгоритми та математичні методи захисту інформації .....	119
Управління виробництвом на базі застосування систем класу ERP .....	120
Оптимізація та балансування навантажень у базах даних .....	120
Програмування смарт-контрактів.....	122
Комп'ютерна лінгвістика .....	123
Системи безпеки програм і даних .....	124
Програмування комп'ютерної графіки.....	125
<b>Освітній компонент 13 Ф-каталогу.....</b>	<b>127</b>
Технології штучного інтелекту.....	127
Інформаційна складова систем радіокерування .....	128
Програмування вбудованих систем на основі архітектури ARM.....	129
Інженерія знань .....	130
Сучасні методи визначення вимог до ІТ-рішень .....	131
Мережеве програмування в середовищі Unix.....	132
Гібридні комп'ютерні системи.....	133
Проектування вбудованих систем .....	134
Технології розроблення вбудованих систем ІоТ (Сертифікатна програма) .....	135
<b>Освітній компонент 14Ф-каталогу.....</b>	<b>139</b>
Інтелектуальні транспортні системи.....	139
Кібербезпека комп'ютерних мереж.....	140
Теорія формальних граматик.....	141
Сучасні технології розробки програмного забезпечення (Авторський курс компанії SoftServe) .....	142
Методика та організація наукових досліджень .....	143
Системи цифрової обробки сигналів і зображень .....	144
Статистичні методи машинного навчання .....	145

## Інструкція користувачам каталогу

1. Кількість вибірових навчальних дисциплін, які може обрати студент визначається навчальним планом. У навчальному плані зазначається також семестр, у якому викладається вибірова дисципліна, форма семестрового контролю –залік, види та обсяги навчальних занять, обсяг (у кредитах ЄКТС).

2.Каталог вибірових дисциплін спільний для студентів усіх спеціальностей та освітніх програм факультету інформатики та обчислювальної техніки(ФІОТ), тому потрібно слугуватися інформацією з розділу «Вимоги до початку вивчення» анотацій та обирати дисципліни відповідно до рівня своєї підготовки.

3. Процедура вибору студентами навчальних дисциплін з Ф-каталогу реалізується через спеціалізовану інформаційну систему Університету [my.kpi.ua](http://my.kpi.ua) та включає такі етапи:

3.1 Реєстрація студента в [my.kpi.ua](http://my.kpi.ua)

3.2 Здійснення студентом вибору дисциплін – перший етап вибору

3.3 У разі неможливості формування навчальних потоків нормативної чисельності для вивчення певної вибірової дисципліни, студентам надається можливість протягом визначеного терміну здійснити повторний вибір, приєднавшись до вже сформованих навчальних потоків – другий етап вибору.

3.4 Підтвердження студенту його вибору навчальних дисциплін із Ф-Каталогу.

3.5 Опрацювання результатів вибору дисциплін (фіксація результатів вибору) та передача даних для корекції індивідуальних навчальних планів аспірантів.

4. За бажанням студента, який обрав певну вибірову дисципліну, допускається його/її приєднання до групи, в якій викладається ця дисципліна в рамках іншої освітньої програми, в тому числі, на іншому факультеті. Відповідне рішення щодо дисциплін, які викладаються на факультеті, ухвалюється деканом факультету. Щодо дисциплін, які викладаються на іншому факультеті (інституті), відповідне рішення ухвалюється деканом факультету за згодою декана того факультету (директора інституту), кафедра якого забезпечує викладання цієї дисципліни.

5. Студент не може двічі обрати одну й ту ж саму навчальну дисципліну, з однаковою назвою. Дисципліну, яку він вже вивчав в попередні курси чи семестр.

6. Якщо студент із поважної причини не зміг обрати дисципліни вчасно, або виявив помилку щодо свого волевиявлення, він звертається в деканат із заявою для запису на вивчення обраних ним дисциплін, надавши відповідні документи. Зміна дисципліни можлива в термін за місяць до початку навчального семестру в якому вона викладається та за наявності вільних місць.

7. Студент, який знехтував своїм правом вибору, буде записаний на вивчення тих дисциплін, які завідувач випускаючої кафедри вважатиме потрібними для оптимізації навчальних груп і потоків.

8. З кожного «Освітнього компоненту Ф-каталогу», обирається одна дисципліна.

9. Всі дисципліни, що представлені в каталозі мають обсяг 4 кредити / 120 годин. Розподіл годин: 54 години аудиторних, з них 36 годин лекцій та 18 годин лабораторних, 66 годин – самостійна робота студентів.

10. Обрані студентом навчальні дисципліни зазначаються у його індивідуальному навчальному плані.

11. Більше інформації про порядок реалізації студентами права на вільний вибір навчальних дисциплін можна знайти у відповідному Положенні про порядок реалізації здобувачами вищої освіти права на вільний вибір навчальних дисциплін.

12. Порядок вибору для студентів, що навчаються по спеціальності 126 «Інформаційні системи та технології» та «121 Інженерія програмного забезпечення» за освітньо-професійними програмами(*групи IA, IC, IK, IP*) *«Інформаційні управляючі системи та технології», «Інформаційне забезпечення робототехнічних систем», «Інтегровані інформаційні системи», «Інженерія програмного забезпечення інформаційних систем»:*

12.1 На 2-й курс, 4-й семестр – обрати один освітній компонент (дисципліну), вибір здійснюється на першому курсі, 2й семестр, серед дисциплін представлених в каталозі як «Освітній компонент 1 Ф-каталогу».

12.2 На 3-й курс, 5-й та 6-й семестри – обрати 7 освітніх компонентів(дисциплін), вибір здійснюється на другому курсі, 4й семестр, серед дисциплін представлених в каталозі як «Освітній компонент 2...8 Ф-каталогу», **окрім** «Освітній компонент 5.1 Ф-каталогу».

12.3 На 4-й курс, 7-й та 8-й семестри – обрати 6 освітніх компонентів(дисциплін), вибір здійснюється на третьому курсі, 6й семестр, серед дисциплін представлених в каталозі як «Освітній компонент 9...14 Ф-каталогу».

13. Порядок вибору для студентів, що навчаються по спеціальності 121 «Інженерія програмного забезпечення» за освітньо-професійною програмою(*групи IM*) *«Інженерія програмного забезпечення комп'ютерних систем»:*

13.1 На 3-й курс, 5-й та 6-й семестри – обрати 8 освітніх компонентів(дисциплін), вибір здійснюється на другому курсі, 4й семестр, серед дисциплін представлених в каталозі як «Освітній компонент 2...8 Ф-каталогу», **в тому числі** і «Освітній компонент 5.1 Ф-каталогу».

13.2 На 4-й курс, 7-й та 8-й семестри – обрати 6 освітніх компонентів(дисциплін), вибір здійснюється на третьому курсі, 6й семестр, серед дисциплін представлених в каталозі як «Освітній компонент 9...14 Ф-каталогу».

14. Порядок вибору для студентів, що навчаються по спеціальності 123 «Комп'ютерна інженерія» за освітньо-професійною програмою(*групи IO*) *«Комп'ютерні системи та мережі»:*

14.1 На 3-й курс, 5-й та 6-й семестри – обрати 7 освітніх компонентів(дисциплін), вибір здійснюється на другому курсі, 4й семестр, серед дисциплін представлених в каталозі як «Освітній компонент 2...7 Ф-каталогу», **в тому числі** і «Освітній компонент 5.1 Ф-каталогу».

14.2 На 4-й курс, 7-й та 8-й семестри – обрати 7 освітніх компонентів(дисциплін), вибір здійснюється на третьому курсі, 6й семестр, серед дисциплін представлених в каталозі як «Освітній компонент 8.1...14 Ф-каталогу».

15. Для залучення до Сертифікатної програми «Інженерія вбудованих систем та Інтернет речей» необхідно окремо обрати кожен дисципліну сертифікатної програми із Ф-каталогу на кожен рік навчання. Інформація про належність дисципліни до сертифікатної програми вказана в описі (анотації) до кожної дисципліни.



# Анотації вибіркових дисциплін

## 2 курс 4 семестр

### Освітній компонент 1Ф-каталогу

Дисципліна	Лінуксopodobні операційні системи
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	2 курс, 4 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисципліни «Операційні системи».
Що буде вивчатися	Архітектура і побудова Linux-подібних операційних систем, головні підсистеми, можливі алгоритми і шляхи реалізації засобів керування ресурсами. Розглядаються методи і механізми розподілу процесорного часу, взаємодії процесів, сумісного доступу до ресурсів, розподілу пам'яті. Вивчаються принципи організації введення-виведення і файлових систем.
Чому це цікаво/треба вивчати	Linux, спершу розроблений для використання окремими ентузіастами на своїх персональних комп'ютерах, пізніше, завдяки підтримці таких компаній, як IBM, Sun Microsystems, HP, Novell та інших, набув неабиякої популярності як серверна операційна система. Linux портований на велику кількість апаратних платформ. Тепер ця ОС досить успішно використовується як на мейнфреймах та суперкомп'ютерах, так і вбудована в багато інших пристроїв (смартфони, планшети ПК, маршрутизатори комп'ютерних мереж (роутери), пристрої автоматизації, системи керування телевізорами та ігровими консолями тощо). Linux все частіше встановлюється і на настільні комп'ютери.
Чому можна навчитися (результати навчання)	Студенти повинні знати: <ul style="list-style-type: none"><li>- основи побудови Linux-подібних операційних систем, їхньої архітектури, вимоги до них, історію їх розвитку і сучасні підходи до їх реалізації;</li><li>- методи і алгоритми керування локальними ресурсами комп'ютера: процесором, пам'яттю, розділюваними ресурсами;</li><li>- принципами реалізації файлових систем;</li><li>- проблеми реалізації мережних функцій операційних систем і способи організації віддаленого виклику процедур і розподілених файлових систем;</li><li>- підходи до реалізації зазначених вище механізмів у сучасних.</li></ul> Студенти повинні уміти: <ul style="list-style-type: none"><li>- користуватись сучасними операційними Linux-системами;</li><li>- формулювати вимоги до операційної системи для вирішення певних прикладних завдань;</li><li>- здійснювати базові налаштування клієнтських операційних систем.</li></ul>
Як можна користуватися набутими знаннями і	Керувати ресурсами обчислювальної системи, взаємодії з прикладним програмним забезпеченням, а також обґрунтовано

уміннями (компетентності)	обирати операційну систему для вирішення певних завдань і грамотно її налаштувати.
Інформаційне забезпечення дисципліни	Силабус дисципліни, опорний конспект лекцій, методичні вказівки до виконання лабораторних робіт, перелік завдань для самостійної роботи студента, перелік питань до МКР та семестрового контролю.
Форма проведення занять	Лекції, лабораторні роботи.
Семестровий контроль	Залік
Викладач	к.т.н., доцент Сокульський О.Є.

Дисципліна	Емпіричні методи програмної інженерії
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	2 курс, 4 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з «Вища математика», «Дискретна математика», «Теорія ймовірностей», «Основи програмування».
Що буде вивчатися	Методи та засоби емпіричних досліджень, актуальних в аспекті ПЗ, зокрема: методи вимірювання емпіричної інформації, статистична обробка та аналіз даних, дискретні й неперервні теоретичні розподіли випадкових величин (вимірюваних ознак), способи статистичного оцінювання їхніх параметрів, перевірка статистичних гіпотез, математичне дослідження типів емпіричних залежностей між вимірюваними величинами, основ кореляційного та дисперсійного аналізу, застосування емпіричних методів для аналізу продуктивності та надійності програмних систем тощо.
Чому це цікаво/треба вивчати	– Математичні методи аналізу даних широко використовують при дослідженні різноманітних систем і процесів – природних, технічних, екологічних, економічних, соціальних тощо. З огляду на це формування відповідних знань та навичок є необхідною складовою підготовки фахівців у галузі системних наук і кібернетики, інформатики та багатьох інших галузей знань.
Чому можна навчитися (результати навчання)	Студент повинен ЗНАТИ: - існуючі підходи обробки та аналізу емпіричних даних та основні методи статистичних розрахунків за допомогою сучасних математичних пакетів; ВМІТИ: - підготувати емпіричні дані до обробки, вибрати статистичні методи для аналізу вибірки та провести інтерпретацію одержаних результатів.
Як можна користуватися набутими знаннями і уміннями (компетентності)	– Використовувати можливості обчислювальної техніки при обробці та аналізі даних.
Інформаційне	Силабус дисципліни, опорний конспект лекцій, методичні вказівки

забезпечення	до виконання лабораторних робіт, перелік завдань для самостійної роботи студента, перелік питань до МКР та семестрового контролю.
Форма проведення занять	Лекції, лабораторні роботи.
Семестровий контроль	Залік
Викладач	к.т.н., доцент Сокульський О.Є.

Дисципліна	Основи клієнтської розробки
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	2 курс, 4 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: основи програмування
Що буде вивчатися	Принципи роботи систем контролю версій (СКВ), колективна робота над проектом використовуючи СКВ, основні принципів побудови веб-сторінок за допомогою мови розмітки HTML, оформлення зовнішнього вигляду сторінок з застосуванням каскадних таблиць стилівCSS; мова JavaScript, управління елементами об'єктної моделі документа DOM та об'єктної моделі браузера BOM за допомогою JavaScript
Чому це цікаво/треба вивчати	Розробка Web-застосунків актуальна на зараз, застосування JavaScript для розробки сайтів та додатків на платформі Node.JS
Чому можна навчитися(результати навчання)	Працювати з проектами використовуючи СКВ, проектувати, розробляти клієнтські Web-застосунки з використанням мов HTML, CCS, JavaScript
Як можна користуватися набутими знаннями і уміннями (компетентності)	Набуті знання можна використовувати в Front-endрозробці, розробці сайтів та додатків на платформі Node JS
Інформаційне забезпечення	Силабус, конспект лекцій, методичні вказівки до виконання лабораторних робіт, презентаційні матеріали
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	ст. викладач Хмелюк М.С.

Дисципліна	Прикладні задачі машинного навчання
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	2 курс, 4 семестр
Обсяг	4 кредити, 120 годин

Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з: математичний аналіз, лінійна алгебра, дискретної математики, будь-якої мови програмування (бажано базові знання з Python), розуміння ООП.
Що буде вивчатися	При проходженні даної дисципліни, студенти познайомляться з технологіями машинного навчання, з термінологією та основними алгоритмами машинного навчання для класифікації, зрозуміють важливість попередньої обробки даних, навчатися працювати з непоміченими даними (кластерний аналіз) та нейронними мережами (глибокими згортковими, рекурентними). На лабораторних заняттях навчатися застосовувати алгоритми AdaBoost за допомогою scikit-learn, виконувати розпізнавання зображень, реалізуєте штучну нейронну мережу з нуля (за допомогою бібліотек TensorFlow, Keras) на мові Python.
Чому це цікаво/треба вивчати	Технології машинного навчання оточують нас, на сьогоднішній день розуміння алгоритмів та застосування їх на практиці є важливим для кожної сучасної людини, зокрема для розуміння правил безпеки персональних даних в Інтернеті.
Чому можна навчитися (результати навчання)	Сучасним методам машинного навчання та вмінню застосовувати технології та алгоритми машинного навчання при розробці програмне забезпечення для різноманітних практичних задач.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Уміння розробляти програмне забезпечення з застосуванням машинного навчання є однією з вимог ІТ-компаній до своїх майбутніх працівників. Отримані знання та уміння в даному курсі можуть стати підґрунтям для обрання теми бакалаврської або магістерської роботи, для подальших наукових досліджень.
Інформаційне забезпечення	Силабус, конспект лекцій (навчальний посібник), навчальний посібник до виконання лабораторних робіт, завдання до виконання лабораторних робіт, перелік питань і завдань для проведення поточного і підсумкового контролю, завдання для комплексної контрольної роботи, завдання до самостійної роботи студентів та методичні вказівки до їх виконання.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	д.т.н., професор Онищенко В.В.

Дисципліна	<b>Вступ до розробки програмного забезпечення</b>
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	<b><u>Окрім студентів, що навчаються по 121 спеціальності.</u></b> Кількість студентів на курс від 60 до 120
Курс, семестр	2 курс, 4 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: Теорія алгоритмів, Основи програмування
Що буде вивчатися	Основні етапи розробки програмного забезпечення. Їх особливості,

	послідовність проходження. Ознайомлення с такими поняттями, як мета, концепція програмного проєкта, предметна область, потреби користувача, вимоги та функціонал, а також їх взаємозв'язок.
<b>Чому це цікаво/треба вивчати</b>	Ця інформація дозволяє отримати перші практичні навички та теоретичні знання для розробки успішних програмних продуктів.
<b>Чому можна навчитися (результати навчання)</b>	Навчитися проводити аналіз предметної області, концепції, мети, потреб користувача, і на їх основі визначення основних вимог програмного забезпечення для нескладних інформаційних систем.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Здатність до абстрактного мислення, аналізу та синтезу Здатність застосовувати знання у практичних ситуаціях Здатність до пошуку, оброблення та узагальнення інформації з різних джерел. Здатність аналізувати об'єкт проєктування або функціонування та його предметну область.
<b>Інформаційне забезпечення</b>	Силабус, конспект лекцій, вказівки до виконання комп'ютерних практикумів, презентації.
<b>Форма проведення занять</b>	Лекції, комп'ютерні практикуми
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., доцент Солдатова М.О.

<b>Дисципліна</b>	<b>Основи розробки програмного забезпечення на платформі Microsoft.NET</b>
<b>Кафедра</b>	Інформатики та програмної інженерії
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	2 курс, 4 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	знання алгоритмів та структур даних, підходів до програмування (наприклад, об'єктно-орієнтоване програмування)
<b>Що буде вивчатися</b>	<ul style="list-style-type: none"> <li>• мова програмування C#;</li> <li>• підходи до побудови програми (принципи об'єктно-орієнтованого проєктування - SOLID, шаблони проєктування – GoF, рекомендації щодо стилю написання програм)</li> <li>• механізм роботи LINQ</li> <li>• основні підходи для роботи з XML в C#.</li> </ul> технології роботи з базами даних ADO.NET.
<b>Чому це цікаво/треба вивчати</b>	отримані знання та набуті практичні навички дозволять виробити певне мислення до процесу розробки програм на мові C#; писати «чистий код» та створювати «читабельне» програмне забезпечення; розробляти програми, які легко супроводжувати, модифікувати, розширювати
<b>Чому можна навчитися (результати навчання)</b>	ефективно використовувати базові конструкції мови програмування C# при створенні програм; правильно застосовувати різні механізми мови C#; правильно обирати та застосовувати колекції при розробці програмного забезпечення для обробки наборів даних; правильно обирати та застосовувати шаблони проєктування для будови

	архітектури програми
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> <li>• використовувати мову програмування C# для розробки програмного забезпечення;</li> <li>• застосувати архітектурні, алгоритмічні та програмні рішення для розробки програмного забезпечення;</li> </ul> використовувати шаблони проектування для створення об'єктів та обробки запитів від користувача
Інформаційне забезпечення	Силабус дисципліни, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	к.т.н., доцент Ліщук К.І.

Дисципліна	Основи розробки програмного забезпечення на платформі Java
Кафедра	Інформатики та програмної інженерії
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 90
Курс, семестр	2 курс, 4 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	знання алгоритмів та структур даних, підходів до програмування (наприклад, об'єктно-орієнтоване програмування), мови програмування Сі або Сі++
Що буде вивчатися	<ul style="list-style-type: none"> <li>• архітектура JVM (розподіл пам'яті);</li> <li>• мова програмування Java (основні конструкції, масиви, реалізація принципів об'єктно-орієнтованого програмування, абстрактні класи чи інтерфейси);</li> <li>• технологія програмування на мові Java - Standard Edition (вкладені класи, generics, колекції, обробка виключень, система введення/виведення даних, логування роботи програми, багатопоточність);</li> </ul> підходи до побудови програми (архітектурний шаблон MVC, принципи об'єктно-орієнтованого проектування - SOLID, шаблони проектування – GoF, рекомендації щодо стилю написання програм)
Чому це цікаво/треба вивчати	отримані знання та набуті практичні навички дозволять виробити певне мислення до процесу розробки програм на мові Java; писати «чистий код» та створювати «читабельне» програмне забезпечення; розробляти програми, які легко супроводжувати, модифікувати, розширювати
Чому можна навчитися (результати навчання)	ефективно використовувати базові конструкції мови програмування Java при створенні програм; правильно застосовувати різні механізми мови Java; правильно обирати та застосовувати колекції при розробці програмного забезпечення для обробки наборів даних; правильно обирати та застосовувати шаблони проектування для будови архітектури програми; створювати багато поточні програми
Як можна користуватися	<ul style="list-style-type: none"> <li>• писати програми у стилі Java за технологією Standard Edition;</li> <li>• налаштовувати систему логування та обробку виключень;</li> </ul>

набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> <li>• обирати способи збереження та обробки наборів даних;</li> <li>• налаштовувати багато поточне програми;</li> </ul> будувати архітектуру програми за MVC шаблоном та використовувати шаблони проектування для створення об'єктів та для обробки запитів від користувача
Інформаційне забезпечення	Силабус дисципліни, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	к.т.н., доцент Іванова Л.М.

Дисципліна	Основи розробки програмного забезпечення мовою програмування PHP
Кафедра	Інформатики та програмної інженерії
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	2 курс, 4 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	знання алгоритмів та структур даних, підходів до програмування (наприклад, об'єктно-орієнтоване програмування)
Що буде вивчатися	<ul style="list-style-type: none"> <li>• мова програмування PHP та технологія програмування на PHP для створення серверних, системних та веб-додатків;</li> <li>• підходи до побудови програми (принципи об'єктно-орієнтованого проектування, шаблони проектування, рекомендації щодо стилю написання програм);</li> <li>• робота з форматами обміну даними та побудова АПІ;</li> <li>• робота з базами даних MySQL та PostgreSQL.</li> </ul>
Чому це цікаво/треба вивчати	Станом на січень 2024 року веб на 76.6% складається із PHP-програм ( <a href="https://w3techs.com/technologies/history_overview/programming_language/ms/y">https://w3techs.com/technologies/history_overview/programming_language/ms/y</a> ), і попит на PHP-розробників як поодиночці, так і у вигляді учасників найбільшої спільноти інтернет-програмістів на PHP зберігається і в подальшому зберігатиметься на високому рівні. Отримані знання та набуті практичні навички дозволять майбутньому PHP-розробнику освоїти типові підходи до розроблення серверних, системних та веб-застосунків, виробити певне мислення до процесу розробки програм мовою PHP; писати «чистий код» та створювати «читабельне» програмне забезпечення; розробляти програми, які легко супроводжувати, модифікувати, розширювати.
Чому можна навчитися (результати навчання)	Ефективно використовувати базові конструкції мови програмування PHP при створенні програм; правильно застосовувати різні механізми мови PHP; правильно обирати та застосовувати структури даних при розробці програмного забезпечення для обробки наборів даних; ефективно використовувати класичне поєднання основних веб-технологій; правильно обирати та застосовувати шаблони проектування для будови архітектури програми; створювати

	інтерактивні веб-сервіси
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Дисципліна надає набір вмінь та компетенцій, що є мінімально необхідними для студента, який бажає в майбутньому стати РНР-розробником програмного забезпечення, а саме: <ul style="list-style-type: none"> <li>• використовувати мову програмування РНР для розробки програмного забезпечення, серверних, системних та веб-додатків;</li> <li>• застосовувати архітектурні, алгоритмічні та програмні рішення для розробки програмного забезпечення;</li> <li>• використовувати шаблони проектування для створення об'єктів та обробки запитів від користувача.</li> </ul>
<b>Інформаційне забезпечення дисципліни</b>	Силабус дисципліни, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
<b>Форма проведення занять</b>	Лекції, лабораторні роботи.
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	ст. викл. Ковтунець О.В.

<b>Дисципліна</b>	<b>Основи розроблення програмного забезпечення на платформі Node.js</b>
<b>Кафедра</b>	Інформаційних систем та технологій
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 150
<b>Курс, семестр</b>	2 курс, 4 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з Основ програмування, Баз даних
<b>Що буде вивчатися</b>	Використання мови JavaScript та платформи Node.js для створення серверної частини веб-застосунків та веб-сервісів
<b>Чому це цікаво/треба вивчати</b>	<ol style="list-style-type: none"> <li>1. Демонструє типові підходи до розроблення веб-застосунків та веб-сервісів на платформі Node.js.</li> <li>2. Дає можливість в майбутньому як самостійно створювати невеликі стартап-проекти так і працювати в великих ІТ-компаніях.</li> <li>3. Node.js є однією з найбільш популярних платформ для розроблення серверної складової на мові JavaScript на даний час.</li> </ol>
<b>Чому можна навчитися (результати навчання)</b>	<ol style="list-style-type: none"> <li>1. Синтаксис та основні бібліотеки мови JavaScript.</li> <li>2. Принципи асинхронної багатозадачності у Node.js.</li> <li>3. Використання фреймворку Express.</li> <li>4. Робота з базами даних та об'єктно-реляційне відображення у застосунках на платформі Node.js.</li> </ol>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<ol style="list-style-type: none"> <li>1. Створювати модулі на платформі Node.js.</li> <li>2. Створювати веб-застосунки та веб-сервіси за допомогою фреймворку Express.</li> <li>3. Розробляти рішення, що інтегруються з зовнішніми системами за допомогою REST API.</li> </ol>
<b>Інформаційне забезпечення</b>	Силабус дисципліни, методичні рекомендації до виконання лабораторних робіт



Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	к.т.н., доцент Букасов М.М.

Дисципліна	Практикум з Linux
Кафедра	Інформатики та програмної інженерії
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	2 курс, 4 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові навички взаємодії з комп'ютером та роботі в інтернеті
Що буде вивчатися	<ul style="list-style-type: none"> <li>• Базові команди взаємодії з операційною системою за допомогою командного рядка (робота с файлами, папками, інсталування стороннього ПО)</li> <li>• Робота с системою контролю версій git</li> <li>• Написання скриптів мовою bash для автоматизації задач різної складності</li> <li>• Основи комп'ютерних мереж, мережева адресація та маршрутизація, активне та пасивне мережеве обладнання.</li> <li>• Налаштування мережевого обладнання.</li> </ul> Взаємодія з АПІ за допомогою командного рядка.
Чому це цікаво/треба вивчати	Отримані знання та набуті навички дозволять зрозуміти основи взаємодії з операційною системою за допомогою командного рядка. Написання bash скриптів дозволять автоматизувати деякі задачі, що виникають в процесі розробки та компіляції проектів. Розуміння мережі дасть можливість правильно реалізовувати комунікацію, наприклад, між клієнтський та серверним програмним забезпеченням.
Чому можна навчитися (результати навчання)	Взаємодіяти с ОС за допомогою командного рядка. Правильно використовувати git в проектах Створювати та використовувати bash для автоматизації процесів різної складності Розуміти основи маршрутизації, без складностей працювати з різним мережевим обладнанням. Працювати з АПІ.
Інформаційне забезпечення	Силабус дисципліни, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	асист. Храмченко М.С.

### 3 курс 5 семестр

#### Освітній компонент 2Ф-каталогу

Дисципліна	Системи масового обслуговування
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін «Математичний аналіз», «Диференціальні рівняння», «Програмування», «Дискретна математика», «Теорія ймовірностей та математична статистика».
Що буде вивчатися	Основні відомості та поняття з теорії масового обслуговування (математична модель системи масового обслуговування, типи систем, вхідні потоки заявок та їх математичні моделі, характеристики систем масового обслуговування та інші) та методів дослідження систем масового обслуговування різних типів.
Чому це цікаво/треба вивчати	У теорії і практиці моделювання систем важливе місце посідають моделі систем масового обслуговування (СМО). Такі системи зустрічаються нам щоденно. Це процеси обслуговування в черзі на заправній станції, у магазині, бібліотеці, кафе, також різні служби ремонту і медичної допомоги, транспортні системи, аеропорти, вокзали тощо. Черги виникають і за потреби скористатись телефонним зв'язком або передати повідомлення по Інтернету. Більше того, будь-яке виробництво також можна подати як послідовність таких систем. Особливого значення СМО набули в інформатиці. Це системи, мережі передавання інформації, бази і банки даних.
Чому можна навчитися (результати навчання)	У результаті вивчення навчальної дисципліни студент повинен <b>знати</b> : основні поняття та методи теорії масового обслуговування <b>вміти</b> : застосовувати засвоєні теоретичні знання при розв'язанні конкретних задач.
Як можна користуватися набутими знаннями і вміннями (компетентності)	Працювати з моделями систем масового обслуговування та застосувати отримані знання до вирішення прикладних задач.
Інформаційне забезпечення	Силабус, опорний конспект лекцій, методичні вказівки до виконання лабораторних робіт, перелік завдань для самостійної роботи студента, перелік питань до МКР та семестрового контролю.
Форма проведення занять	Лекції, лабораторні роботи.
Семестровий контроль	Залік
Викладач	к.т.н., доцент Сокульський О.Є.

Дисципліна	Нечітка логіка
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін «Програмування», «Теорія ймовірностей та математична статистика», «Дискретна математика», «Емпіричні методи програмної інженерії».
Що буде вивчатися	Методита моделі теорії нечітких множин.
Чому це цікаво/треба вивчати	<p>Розвиток нечіткої логіки йде по шляху створення систем, що потрібні великому бізнесу і військовим. Нечітка логіка застосовується при аналізі нових ринків, біржовій грі, оцінці політичних рейтингів, виборі оптимальної цінової стратегії, оцінці рівня зрілості процесів захисту інформації і т.ін. З'явилися і комерційні системи масового застосування.</p> <p>Зсув центра досліджень нечітких систем в бік практичних застосувань привело до постановки цілого ряду проблем, зокрема:</p> <ul style="list-style-type: none"> <li>- нові архітектури комп'ютерів для нечітких обчислень;</li> <li>- елементна база нечітких комп'ютерів і контролерів;</li> <li>- інструментальні засоби розробки;</li> <li>- інженерні методи розрахунку і розробки нечітких систем керування, тощо.</li> </ul>
Чому можна навчитися (результати навчання)	<p>У результаті вивчення навчальної дисципліни студент повинен знати:</p> <ul style="list-style-type: none"> <li>- основні поняття і закони теорії нечітких множин;</li> <li>- основні поняття, методи і прийоми нечіткого аналізу;</li> <li>- основи нечіткої логіки;</li> </ul> <p>вміти:</p> <ul style="list-style-type: none"> <li>- визначати основні об'єкти теорії нечітких множин та виконувати з ними математичні і логічні операції;</li> <li>- застосовувати знання фундаментальних основ, підходи і методи нечіткого аналізу в навчанні та професійній діяльності, в інтегруванні наявних знань, нарощуванні накопичених знань;</li> <li>- застосовувати теоретичні положення теорії нечітких множин та нечіткої логіки в розв'язуванні різноманітних практичних задач.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розв'язуванні різноманітних практичних задач із використанням нечітких моделей та методів нечіткої логіки.
Інформаційне забезпечення	Силабус дисципліни, опорний конспект лекцій, методичні вказівки до виконання лабораторних робіт, перелік завдань для самостійної роботи студента, перелік питань до МКР та семестрового контролю.
Форма проведення занять	Лекції, лабораторні роботи.
Семестровий контроль	Залік
Викладач	к.т.н., доцент Сокульський О.Є.

Дисципліна	Мова програмування Java
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 150
Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з вищої та дискретної математики, процедурного та об'єктно-орієнтованого програмування, теорії алгоритмів.
Що буде вивчатися	Програма курсу передбачає вивчення ядра мови Java: принципів роботи з пам'яттю, типів даних, поглиблене вивчення об'єктно-орієнтованого програмування, робота з класами, методами, виключеннями та узагальненнями, Collections API та Java I/O, лямбда-вирази, багато поточність, рефлексія, інтернаціоналізація, логування, робота з базами даних та графічним інтерфейсом. А також система керування версіями та бібліотеки для тестування. Результатами буде створення простих додатків, заснованих на об'єктно-орієнтованій парадигмі.
Чому це цікаво/треба вивчати	Даний курс орієнтований на вивчення Java як людьми з мінімальним рівнем знання програмування так і людьми, які хочуть покращити свої знання з певних нюансів мови. Після його завершення ви зможете писати програми на Java, і будете мати основу, необхідну для подальшого поглиблення своїх знань та навичок в програмуванні.
Чому можна навчитися (результати навчання)	Особливостям мови програмування Java та основоположним навичкам, необхідним інженеру програмного забезпечення для вирішення проблем у реальному світі, від розробки простих алгоритмів та додатків до тестування та налагодження програм.
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> <li>• проектувати та розробляти пакетні додатки для вирішення різноманітних задач;</li> <li>• користуватися технічною документацією;</li> <li>• проводити тестування додатків за допомогою прикладних бібліотек;</li> <li>• проводити розробку з використанням систем керування версіями;</li> <li>• проводити аналіз побудованих додатків за допомогою програмних засобів.</li> </ul>
Інформаційне забезпечення	Силабус дисципліни, навчально-методичний комплекс.
Форма проведення занять	Лекції та лабораторні роботи
Семестровий контроль	Залік
Викладач	PhD, асистент Орленко С.П.

Дисципліна	Системне програмування С і С++
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: Основ програмування, Бази даних (не критично, можливе одночасне вивчення).
Що буде вивчатися	Синтаксис мов програмування С, С++; операційна система Linux, операційна система (ОС) Windows; середовища розробки QT Creator; основи ядра ОС та його функції; компілятор GCC; розробка драйверів на рівні модулів ядра, бібліотеки QT.
Чому це цікаво/треба вивчати	<ol style="list-style-type: none"> <li>1. Демонструє типовий підхід до розробки програм з використанням мов С, С++ для ОС Windows і Linux, у тому числі на рівні ядра ОС.</li> <li>2. Дає можливість самостійно розробляти прикладні і системні програми.</li> <li>3. Розкриває технології створення графічних інтерфейсів на базі QT.</li> <li>4. Користується попитом у роботодавців, підтримується інформаційно з боку компанії-розробника.</li> <li>5. Виводить студента на початковий професійний рівень.</li> </ol>
Чому можна навчитися (результати навчання)	<ol style="list-style-type: none"> <li>1. Синтаксис і типові патерни (шаблони) мов програмування С, С++.</li> <li>2. Технологію розробки програм на основі мови С з GCC.</li> <li>3. Способи розробки програм на мові програмування С++ з використанням QT Creator для ОС Linux і Windows.</li> <li>4. Типові шаблони для мов програмування С, С++.</li> <li>5. Середовища розробки QT Creator і компілятор GCC.</li> <li>6. Розробка на рівні ядра ОС у тому числі розробка драйверів на рівні модуля ядра.</li> </ol>
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ol style="list-style-type: none"> <li>1. Програмувати і налагоджувати програми на мові С (С++) для розробки драйверів на рівні модулів ядра ОС.</li> <li>2. Розробляти програмне забезпечення на мові С++ для Linux і Windows з використанням QT Creator.</li> <li>3. Створювати прикладні програми і графічні інтерфейси для операційної системи Linux і Windows;</li> <li>4. Налагоджувати і документувати ПЗ.</li> </ol>
Інформаційне забезпечення	Силабус, методичні рекомендації до виконання лабораторних робіт, інформаційна підтримка у Телеграмі:( <a href="https://t.me/SystemProgC_Cplpl">https://t.me/SystemProgC_Cplpl</a> ) група відкривається на час занять) інтегроване середовище розробки QT ( <a href="https://www.qt.io/product/development-tools">https://www.qt.io/product/development-tools</a> ), довідкова система QT ( <a href="https://resources.qt.io/learning-hub">https://resources.qt.io/learning-hub</a> ), компілятор ( <a href="https://gcc.gnu.org">https://gcc.gnu.org</a> ).
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	к.т.н., доц. Ковальов М.О.

Дисципліна	Сучасні технології розробки WEB-застосувань на платформі Microsoft.NET
Кафедра	Інформатики та програмної інженерії
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Обов'язково: програмування – навички процедурного та об'єктно-орієнтованого програмування, наприклад, на мові Сі; знання алгоритмів та структур даних. <u>Бажано</u> : об'єктно-орієнтоване проектування; базове знання С#; реляційні бази даних та SQL; базовий Web – основи HTML та CSS, протокол http.
Що буде вивчатися	Устрій .Net коду та синтаксичні засоби його побудови: узагальнені типи даних, делегати, події, лямбда-вирази та інш. Принципи роботи прибиральника сміття та керування пам'яттю. Технології роботи з даними: ADO.Net, Entity Framework. Багатопотокове та асинхронне програмування та інш.
Чому це цікаво/треба вивчати	Аналіз пропозицій працевлаштування в ІТ-галузі на загальновідомих інтернет-агрегаторах DOU.ua та інш. дає змогу зрозуміти, що розробники програмного забезпечення з використанням технологій Microsoft займають одні з перших позицій щодо попиту на їх знання та вміння, до того ж, якщо є хист та бажання вивчати сучасні мови та технології програмування, С# та .Net є досить молодим та стрімко змінюючимся напрямком створення програмного забезпечення, що надає широкі та цікаві перспективи як у навчанні, так і у подальшому працевлаштуванні
Чому можна навчитися (результати навчання)	Будувати архітектуру програмних систем із компонентів зі слабкою зв'язаністю (багатошарову, мікросервісну, доменно-орієнтовану). Розробляти програмні модулі різного призначення – бізнес-логіка, компоненти доступу до даних, десктопні та web-інтерфейси користувача, компоненти багаторазового використання, розташовані як локально, так і на ресурсах глобального призначення (наприклад, nuget.org), сервіси, розгорнуті у www.
Як можна користуватися набутими знаннями і вміннями (компетентності)	Дисципліна надає набір вмінь та компетенцій, що є мінімально необхідними для студента, який бажає в майбутньому стати С#/ .Net-розробником програмного забезпечення, а саме: <ul style="list-style-type: none"> <li>- створювати програмні продукти на мові С#, що працюють на платформах сімейства .Net,</li> <li>- розробляти бази даних для СУБД MS SQL Server та застосовувати їх у програмних системах .Net,</li> <li>- створювати і розгортати web-сервіси та інші програмні компоненти для використання їх у системах з сервісноорієнтованою та мікросервісною архітектурою,</li> <li>- писати комп'ютерні ігри на платформі unity,</li> <li>- розробляти настільне та web- сучасне програмне забезпечення,</li> <li>- створювати компоненти та інтегрувати їх в існуючі продукти Microsoft: Office, SharePoint</li> </ul>
Інформаційне забезпечення	Силабус дисципліни, лабораторний практикум, презентації лекційного

	матеріалу, контрольні завдання
Форма проведення занять	Лекції та лабораторні (комп'ютерні практикуми)
Семестровий контроль	Залік
Викладач	к.т.н., доцент Крамар Ю.М.

Дисципліна	Сучасні технології розробки WEB-застосувань на платформі Java
Кафедра	Інформатики та програмної інженерії
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 25 до 60
Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	знання мови програмування Java, шаблонів проектування, основ побудови баз даних (БД) та мови запитів SQL, основ веб-програмування – протоколів передачі даних, підходи до процесу розробки програмного забезпечення (життєвий цикл програмного забезпечення)
Що буде вивчатися	<ul style="list-style-type: none"> <li>• підходи до роботи з БД (інтерфейс JDBC, DAO, транзакції, JPA, Hibernate);</li> <li>• побудова веб-застосунків за технологією Enterprise Edition (Servlets, Filters, JSP, JSTL);</li> <li>• локалізацію інтерфейсу користувача;</li> <li>• фреймворк Spring (Spring Data, Spring Core, Spring Security, Spring Rest, Spring Boot);</li> <li>• інструмент автоматичної збірки застосунку – Maven;</li> </ul> фреймворки модульного та інтеграційного тестування – JUnit та Mockito
Чому це цікаво/треба вивчати	отримані знання та набуті практичні навички дозволять розробляти веб-застосунки за фреймворком Spring для будь-якої предметної області з урахуванням географічної локалі; створювати модульні та інтеграційні тести для перевірки працездатності застосунків. Все це необхідно знати та вміти сучасному розробнику застосунків на мові Java
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> <li>• створювати веб-застосунки з використанням різних підходів;</li> <li>• використовувати фреймворк Spring для таких функцій як авторизація та аутентифікація, доступ до даних (БД), організація обробки запитів від користувачів;</li> <li>• створювати слоїсту архітектуру застосунків;</li> <li>• покривати програмне забезпечення застосунків тестовими сценаріями;</li> </ul> локалізувати інтерфейс користувача
Як можна користуватися набутими знаннями і вміннями (компетентності)	<ul style="list-style-type: none"> <li>• будувати архітектуру веб-застосунку з використанням фреймворку Spring;</li> <li>• організувати інтерфейс з підстроюванням до місця розташування користувача;</li> </ul>

	<ul style="list-style-type: none"> <li>налаштовувати Maven для збірки та зв'язку зі сторонніми бібліотеками;</li> </ul> налаштовувати такі функції застосунку як авторизація/аутентифікація, пагінація, транзакція, валідація даних
<b>Інформаційне забезпечення</b>	Силабус дисципліни, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
<b>Форма проведення занять</b>	Лекції та лабораторні (комп'ютерні практикуми)
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., доцент Іванова Л.М.

Дисципліна	Сучасні технології розробки WEB-застосунків мовою програмування PHP
<b>Кафедра</b>	Інформатики та програмної інженерії
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	3 курс, 5 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	<p><u>Обов'язково:</u> програмування – навички процедурного та об'єктно-орієнтованого програмування на основі будь-якої сучасної мови програмування (напр. C, Java, Javascript); знання алгоритмів та структур даних.</p> <p><u>Бажано:</u> об'єктно-орієнтоване проектування; базове знання PHP; реляційні бази даних та SQL; володіння основами вебу – HTML, CSS, Javascript, протокол HTTP/HTTPS.</p>
<b>Що буде вивчатися</b>	<ul style="list-style-type: none"> <li>Особливості мови програмування PHP;</li> <li>підходи до роботи з БД (інтерфейс ODBC, PDO, транзакції, ORM);</li> <li>Використання фреймворків на прикладі Laravel;</li> <li>Техніки забезпечення стійкості програмного забезпечення при високому навантаженні.</li> </ul>
<b>Чому це цікаво/треба вивчати</b>	<p>Станом на січень 2024 року веб на 76.6% складається із PHP-програм (<a href="https://w3techs.com/technologies/history_overview/programming_language/ms/y">https://w3techs.com/technologies/history_overview/programming_language/ms/y</a>), і попит на PHP-розробників як поодиночці, так і у вигляді учасників найбільшої спільноти інтернет-програмістів на PHP зберігається і в подальшому зберігатиметься на високому рівні.</p> <p>Сучасний розробник PHP-застосунків за допомогою отриманих знань та набутих практичних навичок зможе розробляти «читабельні» легко супроводжувані серверні, системні та веб-застосунки на PHP для будь-якої предметної області.</p>
<b>Чому можна навчитися (результати навчання)</b>	<p>Будувати архітектуру програмних систем із компонентів зі слабкою зв'язаністю (багатошарову, мікросервісну, доменно-орієнтовану).</p> <p>Розробляти програмні модулі різного призначення – бізнес-логіка, компоненти доступу до даних, десктопні та web-інтерфейси користувача, компоненти багаторазового використання, розташовані як локально, так і на ресурсах глобального призначення, сервіси,</p>



	розгорнуті у <a href="http://www">www</a> .
Як можна користуватися набутими знаннями і вміннями (компетентності)	<p>Дисципліна надає набір вмінь та компетенцій, що є необхідними для студента, який бажає в майбутньому стати РНР-розробником програмного забезпечення, а саме:</p> <ul style="list-style-type: none"> <li>• створювати програмні продукти мовою РНР, що працюють на усіх доступних платформах;</li> <li>• будувати архітектуру веб-застосунку з використанням фреймворків;</li> <li>• організовувати багатомовний інтерфейс;</li> <li>• розробляти бази даних для різних СУБД та застосовувати їх у програмних системах РНР;</li> <li>• розробляти настільне та веб- сучасне програмне забезпечення;</li> <li>• створювати бібліотеки та інтегрувати їх в популярні існуючі програмні комплекси, написані на РНР;</li> <li>• будувати масштабовані сервіси, що витримують високе навантаження.</li> </ul>
Інформаційне забезпечення	Силабус дисципліни, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
Форма проведення занять	Лекції, лабораторні роботи.
Семестровий контроль	Залік
Викладач	ст. викл. Ковтунець О.В.

Дисципліна	Розробка програмного забезпечення на платформі Node.js
Кафедра	Обчислювальної техніки
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання основ програмування, об'єктно-орієнтованого програмування, алгоритмів та структур даних, комп'ютерних мереж.
Що буде вивчатися	Розробка програм мовою JavaScript. Вивчення основних принципів, базових конструкцій мови. Використання платформи Node.js для розроблення веб-систем. Засоби розробки серверних компонентів та додатків front-end.
Чому це цікаво/треба вивчати	Оволодіння сучасними технологіями розробки веб-систем створення мультиплатформного програмного забезпечення.
Чому можна навчитися (результати навчання)	Створювати ефективно програмне забезпечення веб-систем.
Як можна користуватися набутими знаннями і вміннями (компетентності)	Здатність створювати програмне забезпечення для різноманітних операційних систем та апаратних платформ. Здатність створювати прикладне програмне забезпечення комп'ютерних систем та мереж
Інформаційне	Силабус дисципліни, навчально-методичний комплекс

забезпечення	
Форма проведення занять	Лекції, лабораторні роботи, самостійна робота
Семестровий контроль	Залік
Викладач	ст. викл.Шемсєдинов Т.Г.

Дисципліна	Технології програмування на C/Embedded (Сертифікатна програма)
Кафедра	Обчислювальної техніки
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 90
Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: Дискретна математика, Комп'ютерна логіка, Комп'ютерна арифметика, Програмування, Архітектура комп'ютерів (базові знання), Git (базові знання).
Що буде вивчатися	Дисципліна зосереджена на базових основах розробки програмного забезпечення для застосунків вбудованих систем із використанням мікропроцесорних систем (STM32). Будуть вивчатися: <ul style="list-style-type: none"> <li>- базові основи програмування на мові C;</li> <li>- розвертання та налаштування програмного забезпечення для програмування мікроконтролерів на мові C;</li> <li>- конфігурація та розробка програмного забезпечення для процесорів архітектури ARM;</li> <li>- конфігурація і використання периферії мікроконтролера (GPIO, Timers, PWM, ADC, UART, I2C, SPI);</li> <li>- прошивка фізичних плат та/або емуляторів процесорів на QEMU плат;</li> <li>- використання Git для хостингу проектів та спільної розробки.</li> </ul> Наприкінці дисципліни слухачі розробляють проект з власною прошивкою. Для лабораторних робіт використовуються платформа STM32.
Чому це цікаво/треба вивчати	Вбудовані системи та пристрої IoT потребують програмного забезпечення для функціонування та досягнення своєї робочої мети. Цей курс забезпечує необхідне для початку роботи, а також надає корисні знання для просування в індустрії високих технологій у сфері вбудованих пристроїв та мікроконтролерів. Цей курс розроблено, щоб задовольнити вимоги та потреби студентів, які прагнуть почати програмувати на мові C для вбудованих систем у найлегший і привабливий спосіб.  Галузь застосування. Розроблення рішень для Інтернету Речей (IoT), вбудованих систем, розумних систем. Розроблення апаратних систем для автомобільної промисловості, промислової автоматизації, високопродуктивних обчислень, штучного інтелекту машинного навчання.  Цікаві проекти інноваційних методів навчання вже реалізуються і плануються до використання в курсах «Технології програмування»

	<p>C/Embedded», це – «Автоматизована перевірка лабораторних робіт на Git» та «Лабораторія по віддаленій прошивці плат». Для ініціативних студентів є купа цікавих задачок, де можна проявити себе та отримати сучасний досвід розробки та програмування для вбудованих систем. А компанія GL забезпечила сертифікатну програму базовими комплектами «заліза» для Ваших творчих досягнень.</p> <p>Курс «Технології програмування на C/Embedded» може бути прослуханий студентами як окремий базовий курс, так і у складі <b>сертифікатної програми «Інженерія вбудованих систем та Інтернет речей»</b>, яка запроваджена на факультеті інформатики та обчислювальної техніки з 2021 – 2022 навчального року. Детально з сертифікатною програмою можна ознайомитися на сайті кафедри обчислювальної техніки <a href="https://comsys.kpi.ua/sertifikatni-programi">https://comsys.kpi.ua/sertifikatni-programi</a>. Перелік дисциплін сертифікатної програми:</p> <ul style="list-style-type: none"> <li>• Технології програмування C/Embedded – 5 семестр;</li> <li>• Управління IT-інфраструктурними проектами – 6 семестр;</li> <li>• Технології програмування на ПЛІС (FPGA) – 6 семестр;</li> <li>• Тестування та контроль якості (QA) вбудованих систем – 7 семестр;</li> <li>• Технології розроблення вбудованих систем IoT – 8 семестр.</li> </ul> <p>В цьому курсі і в сертифікатній програмі загалом розглядаються цікаві задачі розроблені у співробітництві з менторами та тренерами компанії GlobalLogic, які також часто приймають участь в лекційних та практичних заняттях. Ми намагаємося допомогти студентам краще зрозуміти реалії сучасного ринку EmbeddedSystems та потреби роботодавців, подолати важкий поріг входження в Embedded та світ HW.</p> <p>Дефіцит фахівців о області EmbeddedSystems та IoT на сьогодні доволі значний, тому опанування дисциплін сертифікатної програми дозволить розширити можливості студентів в пошуку майбутньої роботи та підвищить їх конкурентоспроможність. З точки зору компанії GlobalLogic участь в сертифікатній програмі дозволить сформувати необхідні знання та вміння, що задовільнять її, як роботодавця.</p>
<p><b>Чому можна навчитися (результати навчання)</b></p>	<ul style="list-style-type: none"> <li>– Вивчити основи програмування та основні концепції використання мови програмування C для вбудованих систем;</li> <li>– Отримати стартові навички виконання прошивки фізичних платта/або емуляторів плат на QEMU;</li> <li>– Створювати власні проекти для розроблення прошивки в сучасних середовищах програмування;</li> <li>– Використовувати систему контролю версій кодуGit;</li> <li>– Програмувати на C для процесорів архітектури ARM;</li> <li>– Працювати в команді.</li> </ul>
<p><b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b></p>	<p>Отриманий досвід та практика достатні для того щоб продовжити самостійно професійне навчання в області розроблення нових пристроїв IoT та вбудованих систем з використанням будь якого доступного на ринку апаратури устаткування та програмного забезпечення.</p> <p>Отримані досвід, знання та практика достатні для того, щоб успішно проходити співбесіди в IT компаніях, що займаються розробленням вбудованих систем та IoT.</p>
<p><b>Інформаційне забезпечення</b></p>	<p>Силабус дисципліни, навчально-методичний комплекс.</p>
<p><b>Форма проведення занять</b></p>	<p>Лекції, практичні та семінарські заняття, лабораторні роботи</p>
<p><b>Семестровий</b></p>	<p>Залік</p>

<b>контроль</b>	
<b>Викладач</b>	ассистент Каплунов А.В.

## Освітній компонент ЗФ-каталогу

Дисципліна	Життєвий цикл розробки програмного забезпечення
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 150
Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання основ програмування, базові знання основ системного аналізу
Що буде вивчатися	<ul style="list-style-type: none"> <li>• Окремі питання Бізнес Аналізу при процесі розробки рішень для телеком провайдерів – Requirements engineering, compliance with industry standards (TM Forum, ONF, MEF, 3GPP, IETF тощо)</li> <li>• Окремі питання Software Development при процесі розробки рішень для телеком провайдерів</li> <li>• Окремі питання Application Management при процесі надання сервісу для телеком провайдерів (managed services, change requests implementation and configuration, L1\L2\L3 troubleshooting incident and problem management, change and release management, on-line monitoring and alerting, performance analyses and improvements).</li> </ul>
Чому це цікаво/треба вивчати	Спеціалісти із знаннями в області розробки для телеком провайдерів є особливо затребуваними на ринку праці. Особливо в період впровадження новітніх телекомунікаційних технологій (5G, LEO, IoT тощо)
Чому можна навчитися (результати навчання)	Зрозуміти бізнесові та технічні контексти при розробці програмних рішень для телеком провайдерів
Як можна користуватися набутими знаннями і уміннями (компетентності)	Отримані знання та уміння розширюють перелік потенційних роботодавців для слухачів курсу.
Інформаційне забезпечення	Силабус дисципліни, опорний конспект лекцій (презентації)
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	к.т.н., доц.. Галушко Д.О.

Дисципліна	Методи та технології штучного інтелекту
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 150

Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити ЄКТС
Мова викладання	Українська, англійська
Вимоги до початку вивчення	Базові знання з дисциплін: Алгоритми та методи обчислень, Алгоритми і структури даних, Програмування, Дискретна математика, Інженерія програмного забезпечення
Що буде вивчатися?	Базові знання про контекст, терміни, проблеми та сучасний стан досліджень в галузі штучного інтелекту. Огляд основних принципів і підходів до вивчення систем штучного інтелекту. Огляд основних напрямів досліджень і відповідних інструментів штучного інтелекту: нечіткі системи, нейронні мережі, генетичні алгоритми, еволюційні методи, глибинне навчання, тощо. Огляд основних моделей і алгоритмів із прикладами їх застосування та програмно-апаратними засобами реалізації.
Чому це цікаво/треба вивчати?	Галузь штучного інтелекту останніми роками переживає епоху справжнього піднесення з огляду на бурхливий розвиток нових моделей нейронних мереж, засобів програмного забезпечення та стрімкого розвитку обчислювальних ресурсів на основі графічних (ГПУ) та тензорних (ТПУ) прискорювачів. Дослідження методів штучного інтелекту наразі демонструють найбільш темпи зростання у наукових дослідженнях та широкому практичному застосуванню у багатьох галузях, хоча чого наші життя: від персональних асистентів в смартфонах до самокерованих автівок. Розробка і впровадження систем зі штучним інтелектом дозволяє автоматизувати вирішення задач, що потребують знань. Задач, вирішення яких було не доступне без застосування фахівців в певній галузі.
Чому можна навчитися? (результати навчання)	Теоретичні знання та базовий практичний досвід у застосуванні різних методів штучного інтелекту до існуючих практичних проблем в широкому колі застосувань.
Як можна користуватися набутими знаннями і уміннями? (компетентності)	Здобуті знання дозволять зрозуміти місце і роль методів штучного інтелекту у загальному контексті інформаційних технологій. Ця дисципліна є необхідним етапом для підготовки до професійного рівня оволодіння спеціалізованими знаннями та навичками в галузі штучного інтелекту, які будуть викладатися у наступних навчальних дисциплінах, які присвячені більш детальному вивченню окремих методів штучного інтелекту.
Інформаційне забезпечення	Силабус дисципліни, конспект лекцій, презентаційні матеріали, методичні вказівки до виконання лабораторних робіт
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	к.т.н., доцент Шимкович В.М.

<b>Дисципліна</b>	<b>Технології та засоби розробки комп'ютерної графіки та мультимедіа</b>
<b>Кафедра</b>	Інформаційних систем та технологій

<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	3 курс, 5 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: вища математика, основи програмування
<b>Що буде вивчатися</b>	Методи представлення графічних об'єктів, колірні моделі, двовимірні перетворення, тривимірні перетворення, мультимедіа технології, управління графічними об'єктами мовою C#, використовуючи GDI, OpenGL, програми для обробки мультимедіа інформації та інструментальні засоби створення мультимедіа продукції
<b>Чому це цікаво/треба вивчати</b>	Графіка має широкий спектр областей застосування
<b>Чому можна навчитися (результати навчання)</b>	Проводити аналіз та обирати відповідні до поставленої задачі методи та сучасні програмні середовища, використовувати основні сучасні засоби растрової й векторної графіки, створювати об'єкти двовимірної та тривимірної графіки й анімації, вільно створювати різні зображення за допомогою GDI, OpenGL та комп'ютерних програм; обробляти зв'язувати мультимедіа інформацію
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Набуті знання можна використовувати при розробці програмного забезпечення, створенні мультимедіа продукції
<b>Інформаційне забезпечення</b>	Силабус, конспект лекцій, методичні вказівки до виконання лабораторних робіт, презентаційні матеріали
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	ст. викладач Хмельюк М.С.

<b>Дисципліна</b>	<b>Комп'ютерна графіка та мультимедіа</b>
<b>Кафедра</b>	Інформатики та програмної інженерії
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	3 курс, 5 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	навички програмування на мові високого рівня; знання лінійної алгебри, геометрії та тригонометрії; розв'язування рівнянь, похідних, інтегралів; знання векторів, матриць; вміння розв'язувати системи рівнянь.
<b>Що буде вивчатися</b>	<ul style="list-style-type: none"> <li>• вступ до комп'ютерної графіки;</li> <li>• програмування 2D графіки;</li> <li>• трансформація та відображення об'єктів 3D графіки;</li> <li>• моделювання 3D об'єктів та знаходження видимої поверхні;</li> <li>• освітлення та рендеринг поверхні;</li> </ul> алгоритми трасування променів та їх застосування.

<b>Чому це цікаво/треба вивчати</b>	отримані знання та набуті практичні навички дозволяють розуміти принципи створення та редагування цифрових зображень, а також комп'ютерних ігор.
<b>Чому можна навчитися (результати навчання)</b>	<ul style="list-style-type: none"> <li>• пояснювати основні концепції комп'ютерної графіки, включаючи перегляд, проекцію, перспективу, моделювання та перетворення у двох та трьох вимірах;</li> <li>• застосовувати концепції колірних моделей, моделей освітлення та затінення, текстур, трасування променів, прихованої поверхні, згладжування та рендерингу;</li> <li>• інтерпретувати математичні засади комп'ютерної графіки;</li> <li>• описувати основи анімації, параметричні криві та поверхні, а також точкове освітлення;</li> <li>• визначати типовий графічний конвеєр та застосувати методи графічного програмування для створення комп'ютерної графіки; створювати ефективні програми для вирішення задач програмування комп'ютерної графіки, включаючи 3D-трансформацію, моделювання об'єктів, колірне моделювання, освітлення, текстури та трасування променів.</li> </ul>
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>• створювати цифрові зображення та інтерактивну 3D графіку;</li> <li>• розробляти комп'ютерні ігри та додатки.</li> </ul>
<b>Інформаційне забезпечення</b>	Силабус дисципліни, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
<b>Форма проведення занять</b>	Лекції та лабораторні (комп'ютерні практикуми)
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., доцент Родіонов П.Ю.

<b>Дисципліна</b>	<b>Проектування програмного забезпечення засобами UML</b>
<b>Кафедра</b>	Інформатики та програмної інженерії
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 100
<b>Курс, семестр</b>	3 курс, 5 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Вимоги до програмно-інформаційних систем
<b>Що буде вивчатися</b>	Графічна мова UML
<b>Чому це цікаво/треба вивчати</b>	Графічна мова UML для проограмістів дає змогу представити програмній додаток на формальному рівні та спроектувати його
<b>Чому можна навчитися (результати навчання)</b>	Формальне проектування програмного забезпечення
<b>Як можна</b>	Проектувати програмне забезпечення в загалі та програмні додатки



користуватися набутими знаннями і уміннями (компетентності)	зокрема
Інформаційне забезпечення дисципліни	Силабус дисципліни, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	к.т.н., доц. Новінський В.П.

Дисципліна	Розробка програмного забезпечення на платформі Java
Кафедра	Обчислювальної техніки
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 150
Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання основ програмування, об'єктно-орієнтованого програмування, алгоритмів та структур даних, комп'ютерної дискретної математики
Що буде вивчатися	Принципи побудови мови Java, основні конструкції, бібліотеки. Реалізація принципів об'єктно-орієнтованого програмування, модульність, робота у мережі та з базами даних. Класи, інтерфейси, пакети та інші основні елементи побудови програм. Засоби розробки програм та інтегровані середовища розробки та налагодження програмного забезпечення. Віртуальна машина Java. Організація багатопотокового виконання.
Чому це цікаво/треба вивчати	Оволодіння сучасними Java-технологіями створення платформонезалежного програмного забезпечення, технологіями розробки складних інформаційних систем для вирішення широкого спектру завдань.
Чому можна навчитися (результати навчання)	Створювати ефективне програмне забезпечення на основі технологій Java, вивчити та опанувати сучасні засоби розробки програм на основі об'єктно-орієнтованого підходу та популярних патернів проектування та програмування.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Здатність створювати програмне забезпечення для різноманітних операційних систем та апаратних платформ від мережних до мобільних та вбудованих. Здатність створювати прикладне програмне забезпечення комп'ютерних систем та мереж
Інформаційне забезпечення	Силабус дисципліни, навчально-методичний комплекс
Форма проведення занять	Лекції, лабораторні роботи, самостійна робота
Семестровий контроль	Залік

Викладач	ст. викл. Алещенко О.В.
----------	-------------------------

Дисципліна	Графічне та геометричне моделювання і створення реалістичних зображень
Кафедра	Обчислювальної техніки
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з вищої математики, фізики, аналітичної геометрії, алгоритмів та структур даних, програмування, алгоритмів та структур даних, архітектури комп'ютера
Що буде вивчатися	Моделювання просторових об'єктів і сцен за допомогою сучасних програмних засобів. Методи побудови реалістичних зображень з урахуванням законів оптики. Сучасні апаратно-програмні засоби для моделювання тривимірної реальності.
Чому це цікаво/треба вивчати	Безперервне зростання потужностей комп'ютерних засобів дозволяє використовувати якісну графіку в усе більших застосуваннях. Широко застосовуються реалістичні зображення у кіно, комп'ютерних іграх, архітектурних та геоінформаційних системах тощо. Набувають подальшого розвитку системи віртуальної реальності.
Чому можна навчитися (результати навчання)	Навчитися роботі з популярними програмними засобами комп'ютерної графіки. Вивчити популярні графічні файлові формати. Отримати знання математичних основ комп'ютерної графіки, методам створення реалістичних зображень тривимірних сцен. Вивчити сучасні апаратно-програмні засоби для моделювання тривимірної реальності.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Виконувати тривимірне геометричне моделювання на основі сучасних апаратно-програмних засобів. Знання математичних та алгоритмічних основ комп'ютерної графіки разом із знанням сучасних програмних інтерфейсів графіки відкриває можливість програмісту створювати власні програми, насичені графічними можливостями якісного відображення різноманітних об'єктів та сцен у різноманітних застосуваннях для широкого спектру ІТ-технологій.
Інформаційне забезпечення	Силабус дисципліни, навчально-методичний комплекс
Форма проведення занять	Лекції, лабораторні роботи, самостійна робота
Семестровий контроль	Залік
Викладач	д.т.н., професор Писарчук О.О. ст. викладач Пономаренко А.М.

Дисципліна	Технології Computer Vision
Кафедра	Обчислювальної техніки
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 6 семестр
Обсяг	4 кредити ЄКТС, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з програмування, архітектури обчислювальних систем і мереж, дискретної математики, комп'ютерної логіки, процесів інженерії програмного забезпечення.
Що буде вивчатися	<p>Дисципліна має на меті: надання ґрунтовного теоретичного базису та потужних практичних навичок програмної реалізації методів, математичних моделей і алгоритмів технологій Computer Vision (комп'ютерного бачення = комп'ютерного зору).</p> <p>Теоретичні основи Computer Vision надаються у формі лекційних занять з обов'язковою демонстрацією практичної реалізації розглянутих алгоритмів у формі прикладів програмного коду.</p> <p>Практичні навички застосування технологій Computer Vision набуваються на лабораторних заняттях, які побудовані за принципом нарощування функціональності розроблених скриптів. При цьому приділяється увага процесам інженерії програмного забезпечення.</p> <p>Практична частина дисципліни орієнтована на застосування мови програмування високого рівня Python з вивченням можливостей графічних бібліотек Graphics, Tkinter, Matplotlib, NumPy (для «сирої» реалізації алгоритмів Computer Vision) та спеціалізованих пакетів типу PIL/ Pillow, OpenGL, OpenCV для створення програмних модулів завершеної практичної спрямованості.</p> <p>Дисципліна розкриває суть етапів класичного конвеєра обробки цифрових графічних зображень: синтез зображення (просторове перетворення) – растеризація (реалістичне зображення, цифрова обробка) – векторизація (ідентифікація об'єкту та цільова обробка зображень). Зазначене реалізується у темах:</p> <ol style="list-style-type: none"> <li>1. Синтез цифрового зображення (2D, 3D – об'єкти, графічний метод, аналітичні та неаналітичні методи);</li> <li>2. Базові геометричні перетворення 2D, 3D об'єктів (масштабування, переміщення, обертання, проекції);</li> <li>3. Растеризація і растрові зображення, принципи створення, характеристики та алгоритми обробки (растеризація, зміна яскравості, фільтрація);</li> <li>4. Векторизація і векторні зображення, математичні моделі, алгоритми апроксимації, інтерполяції та згладжування (МНК, Сплайни, Криві Безье тощо);</li> <li>5. Моделі реалістичних зображень, доповнена реальність, моделі світла і тіні: 3D-конвеєр, рендерінг;</li> <li>6. Цифрова обробка зображень: фільтрація; корекція кольору; сегментація; векторизація; морфологічна обробка; розпізнавання.</li> <li>7. Методологічні, технологічні та прикладні аспекти ComputerVision: обробка сенсорних даних; навігація; штучний інтелект; управління.</li> </ol>
Чому це цікаво/треба	Опанування знань, умінь та навичок, що потребують посади: Software Developer with Computer Vision; Embedded developer for

<b>вивчати</b>	Computer Vision systems; Computer Vision Research Engineer тощо.
<b>Чому можна навчитися (результати навчання)</b>	<p>1. Знання та навички застосування і реалізації базових алгоритмів Computer Vision:</p> <ul style="list-style-type: none"> <li>- синтез 2D, 3D графічних об'єктів та їх геометричні перетворення;</li> <li>- синтезу та обробки растрових зображень;</li> <li>- синтезу та обробки векторних зображень;</li> <li>- формування реалістичних зображень;</li> <li>- цифрової обробки зображень;</li> <li>- прикладних аспектів Computer Vision;</li> <li>- комплексної реалізації етапів конвеєру цифрової обробки зображень для задач Computer Vision та інших прикладних галузей.</li> </ul> <p>2. Знання і навички застосування базових бібліотек Python для реалізації алгоритмів Computer Vision: Matplotlib, NumPy PIL/ Pillow, OpenCV.</p>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<p>Набуті компетенції з Computer Vision можливо застосовувати на проєктах наступних прикладних галузей і технологій:</p> <ol style="list-style-type: none"> <li>1. Реконструкція 3D сцени та навігація в ній (Structure-from-Motion, Road Scene Understanding and Autonomous Driving);</li> <li>2. Сегментація сцени і розуміння сюжету (Significance maps, video and 3D segmentation, 3D stream, multiple tracking, object detection, activity detection and prediction, group analysis, object detection and recognition);</li> <li>3. Розпізнавання зображень, ідентифікація об'єктів (Convolution networks, Deep learning, Image Retrieval, Object Detection).</li> </ol>
<b>Інформаційне забезпечення</b>	<p>Силабус дисципліни, навчально-методичний комплекс:</p> <p><a href="https://drive.google.com/drive/folders/1qXc3Ami_HNiVaTPIY7jLom0YRwtiGTQr?usp=sharing">https://drive.google.com/drive/folders/1qXc3Ami_HNiVaTPIY7jLom0YRwtiGTQr?usp=sharing</a><a href="https://classroom.google.com/c/NDIxODIzNTEyNDc2?cjc=mgwrbui">https://classroom.google.com/c/NDIxODIzNTEyNDc2?cjc=mgwrbui</a></p>
<b>Семестровий контроль</b>	Залік, 3 курс 6 семестр
<b>Викладач</b>	д.т.н., професор Писарчук О.О.

## Освітній компонент 4Ф-каталогу

Дисципліна	Технології просування сайтів в пошукових системах
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс	3 курс, 5 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	українська
Вимоги до початку вивчення	Базові знання створення сайтів.
Що буде вивчатися	<p><b>Види інтернет додатків:</b> Landing сторінка, інформаційний сайт, інтернет-магазин, корпоративний сайт, портал.</p> <p><b>Життєвий цикл розробки web додатків.</b></p> <p><b>Створення адаптивного дизайну сайту:</b> Usability, ProgressiveEnhancement, Responsivedesign (MediaQueiries, FlexibleMedia, Fluidgrid).</p> <p><b>Хостинг сайту:</b> віртуальний хостинг, віртуальний виділений сервер (VPS / VDS), виділений сервер, colocation.</p> <p><b>Просування сайтів:</b> GoogleAds, SMM, SMO, SMP, SEO (створення семантичного ядра, стратегії SEO оптимізації, внутрішня та технічна оптимізація сайту, внутрішня перелінковка, зовнішні чинники, інструменти SEO оптимізації).</p>
Чому це цікаво/треба вивчати	Сучасний світ характеризується стрімким розвитком і впровадженням інтернет додатків. Практично кожна компанія повинна мати власний сайт. Тому вміння просувати сайти в пошукових системах є доволі актуальним.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> <li>▪ Розробляти адаптивні макети сайтів;</li> <li>▪ Просувати сайт в пошукових системах за допомогою SEO оптимізації та технології GoogleAds.</li> <li>▪ Просувати сайт за допомогою SEO оптимізації та технології SMM, SMO, SMP.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> <li>▪ для просування Landing сторінок;</li> <li>▪ для просування інформаційних сайтів та інтернет - магазинів;</li> <li>▪ для просування корпоративних сайтів.</li> </ul>
Інформаційне забезпечення	Силабус дисципліни, навчально-методичний комплекс.
Форма проведення занять	Лекції, практичні заняття.
Семестровий контроль	Залік
Викладач	к.т.н., доцент Крилов Є.В.

Дисципліна	Основи Front-end технологій
Кафедра	Інформаційних систем та технологій

Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 150
Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: Базові знання мов програмування, основ графіки та дизайну, роботи в мережі інтернет.
Що буде вивчатися	Прототип сайту ( <b>Moqups, Draftium</b> ), контроль версій <b>GitHub, HTML5</b> та <b>CSS3</b> (basic, advanced), <b>JavaScript</b> (basic, advanced), <b>React.js</b> . Розробка елементів сайту. Верстка сайту за допомогою HTML-5, CSS-3. Адаптивна верстка сайту. Розробка інтерфейсу користувача за допомогою JS та React.js.
Чому це цікаво/треба вивчати	Front end – це один із найпопулярніших напрямків у ІТ. Front-end developers є невід'ємними учасниками будь-якої команди, яка займається розробкою веб-сайтів, веб-застосунків або будь-яких продуктів у WEB.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> <li>▪ розробляти прототипи сайтів;</li> <li>▪ розробляти адаптивні макети сайтів;</li> <li>▪ розробляти адаптивну верстку сайту;</li> </ul> розробляти динамічні сайти за допомогою JS та React.js;
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> <li>▪ для розробки статичних сторінок;</li> <li>▪ для розробки адаптивних WEB-сторінок;</li> <li>▪ для розробки динамічних WEB-сторінок</li> <li>▪ для розробки інтерфейсу користувача за допомогою JS.</li> </ul>
Інформаційне забезпечення	Силабус дисципліни, навчально-методичний комплекс, VSCode, GitHub, Moqups, Draftium, HTML5, CSS3, JS, React.js
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	к.т.н., доцент Жереб К.А.

<b>Дисципліна</b>	<b>Розроблення застосунків з використанням SpringFramework</b>
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 150
Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з Основ програмування, ООП, Веб-програмування, Баз даних
Що буде вивчатися	Використання мови Java та Spring Framework для створення веб-застосунків та веб-сервісів
Чому це цікаво/треба вивчати	<ol style="list-style-type: none"> <li>1. Демонструє типові підходи до розроблення веб-застосунків та веб-сервісів.</li> <li>2. Дає можливість в майбутньому як самостійно створювати невеликі</li> </ol>

	стартап-проекти так і працювати в великих ІТ-компаніях. 3. Spring Framework є одним з найбільш популярних фреймворків для розроблення застосунків для платформи Java на даний час.
<b>Чому можна навчитися (результати навчання)</b>	1. Керування залежностями між компонентами на основі принципу Inversion of Control. 2. Створення веб-застосунків та веб-сервісів на основі Spring MVC. 3. Робота з базами даних за допомогою Spring Data (JDBC, JPA). 4. Декларативне керування безпекою та транзакціями в застосунках Spring Framework. 5. Створення та конфігурування застосунків Spring Framework за допомогою Spring Boot.
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	1. Створювати веб-застосунки та веб-сервіси з використанням Spring Framework. 2. Проектувати та розробляти компоненти для ентєрпрайз-рішень з мікросервісною архітектурою. 3. Розробляти рішення, що інтегруються з зовнішніми системами.
<b>Інформаційне забезпечення</b>	Силабус дисципліни, методичні рекомендації до виконання лабораторних робіт
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., доцент Букасов М.М.

<b>Дисципліна</b>	<b>Технологія блокчейн</b>
<b>Кафедра</b>	Інформаційних систем та технологій
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	3 курс, 5 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з <i>мов та технологій програмування, кодування та шифрування інформації, бази даних, мережеві технології</i>
<b>Що буде вивчатися</b>	Підходи, засоби та технології програмування розподілених баз даних типу блокчейн. Теоретичні засади щодо архітектури, механіки роботи систем на базі блокчейн. Токеноміка та цифровізація послуг, монетизація творчості та рейтингування соціуму, суверенна особистість та децентралізовані сертифікати.
<b>Чому це цікаво/треба вивчати</b>	– Тотальна цифровізація та діджиталізація бізнес-процесів продукують запити до нових компетенцій серед фахівців з інформаційних систем. Зокрема, існує дефіцит у фахівцях з децентралізованих баз даних блокчейн, спеціалістів зі сфери цифрових/віртуальних активів, смарт-контрактів та децентралізованих фінансів.
<b>Чому можна навчитися (результати навчання)</b>	Базові підходи, методи та технології розгортання прототипу розподіленої бази даних типу блокчейн задля підвищення якості бізнес-процесів в інформаційних системах.
<b>Як можна користуватися</b>	– Допомогати у розгортанні та супроводженні розподілених блокчейн-платформ й монетизації бізнес-процесів

набутими знаннями і уміннями (компетентності)	
Інформаційне забезпечення	Силабус дисципліни, посібник до виконання лабораторних робіт, конспект лекцій, дистанційний курс
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	ст. викладач Яланецький В.А.

Дисципліна	Linux
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: операційні системи
Що буде вивчатися	Структура системи, конфігурування системи, командний рядок та його використання для роботи зі стандартними потоками, текстовими редакторами, управлінням файлами, каталогами, правами доступу, користувачами та групами, процесами, журналюванням операцій. Команди мови Shell, скрипти. Планування та планувальники задач в Linux, управління пакетами Linux, управління апаратним забезпеченням в Linux. Проектування, монтування файлових систем. Налаштування мережевих параметрів в Linux. Налаштування графічного середовища
Чому це цікаво/треба вивчати	Linux-системи використовуються на ПК, серверах
Чому можна навчитися (результати навчання)	Працювати з системою, використовуючи командний рядок Shell; управляти файлами, каталогами, правами доступу, користувачами та групами, процесами, журналюванням операцій в Linux; конфігурувати систему через налаштування пакетів, апаратного забезпечення, файлових систем, графічного середовища, мережі.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Набуті знання можна використовувати адміністрування Linux-систем
Інформаційне забезпечення	Силабус дисципліни, конспект лекцій, методичні вказівки до виконання лабораторних робіт, презентаційні матеріали
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	залік
Викладач	Старший викладач Хмелюк М.С.



Дисципліна	Розробка мобільних застосунків під iOS
Кафедра	Інформатики та програмної інженерії
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	знання мови програмування C# та XML, основ побудови баз даних (БД) та мови запитів SQL, основ веб-програмування, алгоритмів та структур даних
Що буде вивчатися	архітектура операційної системи iOS та структури застосунку під неї; мова програмування Swift; розробка нативних застосунків під iOS; взаємодія з базами даних та файловою системою; доступ до глобальної мережі та розробка веб-застосунку; організація роботи з мультимедіа, датчиками, анімацією
Чому це цікаво/треба вивчати	отримані знання та набуті практичні навички дозволять розробляти мобільні застосунки під iOS як нативні, так і гібридні застосунки для будь-якої предметної області
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> <li>розробляти нативні застосунки під iOS;</li> <li>розробляти гібридні застосунки під iOS;</li> <li>організувати сховища для збереження даних у пам'яті пристрою або у БД;</li> </ul> використовувати інтернет-джерела для доступу к даним
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> <li>аналізувати та обирати тип мобільного застосунку;</li> <li>розробляти мобільні версії сайтів;</li> <li>застосовувати різні інтерактивні елементи для зручного інтерфейсу мобільного застосунку;</li> </ul> аналізувати та обирати сховища для збереження даних з урахуванням безпеки по доступу
Інформаційне забезпечення	Силабус дисципліни, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
Форма проведення занять	Лекції та лабораторні (комп'ютерні практикуми)
Семестровий контроль	Залік
Викладач	асист. Храмченко М.С.

Дисципліна	AGILE методологія розробки програмного забезпечення (Авторський курс компанії SoftServe)
Кафедра	Обчислювальної техніки
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 150
Курс, семестр	4 курс, 7 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська

<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: Інженерія програмного забезпечення, Програмування, Об'єктно-орієнтоване програмування, Алгоритми та методи обчислень, Системне програмування, Алгоритми і структури даних, Архітектура комп'ютерів
<b>Що буде вивчатися</b>	основи термінології і основних понять. Основні методології розробки ПО. Методи вибору технологій для розробки. Огляд існуючих технологічних рішень які використовуються для створення проєктивоснови застосування теоретичних знань на прикладі розробки документації навчального проєкту персоналізованого для кожного студента чи групи студентів за їх вибором
<b>Чому це цікаво/треба вивчати</b>	Існуючі потреби та тенденції на ринку праці показують що для успішної роботи програмістом потрібно не тільки вміти писати код (програмувати) також потрібне розуміння принципів розробки ПО в цілому. До 50% часу технічної співбесіди витрачається саме на питання які стосуються методологій розробки оскільки домінуючою методологією розробки ПО на даний час є AGILE отримати розуміння та практику розробки власного проєкту з 0 дозволить достатньо підготуватися до майбутньої роботи
<b>Чому можна навчитися (результати навчання)</b>	Розібратися в реальному процесі розробки ПО. Підготуватися до того що буде очікувати студента після працевлаштування на реальне робоче місце.
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	Розуміти як читати та писати проєктну документацію для проєкта в цілому чи його окремих компонент. Обирати методології та технології для реалізації проєкта чи його компонент.
<b>Інформаційне забезпечення</b>	Силабус дисципліни, навчально-методичний комплекс.
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	Ст. викладач Шевело Олексій Павлович

## Освітній компонент 5.1Ф-каталогу (для кафедри ОТ)

Дисципліна	Вступ до штучного інтелекту
Кафедра	Обчислювальної техніки
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 100
Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська, англійська
Вимоги до початку вивчення	Базові знання з дисциплін: Алгоритми та методи обчислень, Системне програмування, Системне програмне забезпечення, Алгоритми і структури даних, Організація баз даних, Програмування, Архітектура комп'ютерів, Комп'ютерні мережі, Дискретна математика, Інженерія програмного забезпечення, Паралельне програмування
Що буде вивчатися	Базові знання про контекст, терміни, проблеми та сучасний стан досліджень в галузі штучного інтелекту. Огляд основних принципів і підходів до вивчення систем штучного інтелекту. Огляд основних напрямів досліджень і відповідних інструментів штучного інтелекту: генетичні алгоритми, еволюційні методи, машинне навчання, глибинне навчання, навчання із підкріпленням, тощо. Огляд основних моделей і алгоритмів із прикладами їх застосування.
Чому це цікаво/треба вивчати	Галузь штучного інтелекту останніми роками переживає епоху справжнього піднесення з огляду на бурхливий розвиток нових моделей нейронних мереж, засобів програмного забезпечення та стрімкого розвитку обчислювальних ресурсів на основі графічних (ГПУ) та тензорних (ТПУ) прискорювачів. Дослідження методів штучного інтелекту наразі демонструють найбільш темпи зростання у наукових дослідженнях та широкому практичному застосуванню у багатьох галузях оточуючого нас життя: від персональних асистентів в смартфонах до самокерованих автівок.
Чому можна навчитися (результати навчання)	Теоретичні знання та базовий практичний досвід у застосуванні різних методів штучного інтелекту до існуючих практичних проблем в широкому колі застосувань.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Здобуті знання дозволять зрозуміти місце і роль методів штучного інтелекту у загальному контексті інформаційних технологій. Ця дисципліна є необхідним етапом для підготовки до професійного рівня оволодіння спеціалізованими знаннями та навичками в галузі штучного інтелекту, які будуть викладатися у наступних навчальних дисциплінах, які присвячені більш детальному вивченню окремих методів штучного інтелекту.
Інформаційне забезпечення	Силабус дисципліни, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	Старший викладач кафедри ОТ Таран В.І.

Дисципліна	Технології програмування користувацьких інтерфейсів (Front-end)
Кафедра	Обчислювальної техніки
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 100
Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	<p>Базові знання з дисциплін:          Алгоритми та методи обчислень, Системне програмування, Системне програмне забезпечення, Алгоритми і структури даних, Організація баз даних, Вступ до операційної системи Linux, Програмування, Архітектура комп'ютерів, Комп'ютерні мережі, Дискретна математика, Комп'ютерна логіка, Інженерія програмного забезпечення, Комп'ютерні системи</p>
Що буде вивчатися	<p>Дисципліна має на меті: надання ґрунтовних теоретичних знань і конкретних практичних навичок розробки Front-end (клієнтського) програмного забезпечення.</p> <p>Навчальна дисципліна спрямована на формування компетентцій з розробки Front-end частини прикладних Web-застосунків. В поєднанні із навчальною дисципліною «Технології розроблення серверного програмного забезпечення (Back-end)» (7 семестр) забезпечується набуття комплексних знань і практичних навичок проектування і розробки прикладного програмного забезпечення, що базується на сучасних і перспективних клієнт-серверних технологіях розподіленої обробки і обміну інформацією.</p> <p>Теоретична і практична складові навчальної дисципліни спрямовані на набуття компетенцій у ході реалізації конкретного проекту з розробки Front-end частини із спрощеною реалізацією Back-end складової.</p> <p>Дисципліна складається із наступних блоків:</p> <ol style="list-style-type: none"> <li>1. Прототипування інтерфейсу програмного додатку (основи UI/UX дизайну; принципи розробки прототипів; DOM-дерево, фреймворк Bootstrap та його аналоги. LESS та SASS препроцесори);</li> <li>2. Технології програмування користувацьких інтерфейсів (HTML5; CSS5; JavaScript; TypeScript; фреймворк jQuery; фреймворк React та архітектура Flux; фреймворк Angular 6 та архітектура MVC, VueJS);</li> <li>3. Angular 6 (angular CLI, архітектура, модуль, контролер, директива, вираз; фільтри; моделі і валідатори; залежності та сервіси).</li> </ol>
Чому це цікаво/треба вивчати	Опанування знань, умінь та навичок, що потребують посади: Front-end Developer, Front-end Engineer, Front-end Team Lead, DevOps Engineer тощо.
Чому можна навчитися (результати навчання)	<ol style="list-style-type: none"> <li>1. Знання та навички застосування і реалізації клієнт-сервісних технологій для створення ефективних Front-end частин сучасних Web застосунків:             <ul style="list-style-type: none"> <li>прототипування інтерфейсу Front-end;</li> <li>технології програмування користувацьких інтерфейсів.</li> </ul> </li> <li>2. Знання і навички застосування базових Front-end – технологій: HTML5; CSS3; JavaScript; TypeScript; фреймворк jQuery; фреймворк React та архітектура Flux; фреймворк Angular 6 та архітектура MVC.</li> </ol>

Як можна користуватися набутими знаннями і уміннями (компетентності)	Набуті компетенції з розроблення Front-end програмного забезпечення можливо застосовувати на проектах наступних прикладних галузей і технологій: 1. Інтернет-комерція; 2. Електронний документообіг; 3. CRM-системи різного прикладного спрямування, побудованих на технологіях розподілених процесах збору, зберігання, обробки і обміну інформацією.
Інформаційне забезпечення	Силабус дисципліни, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи.
Семестровий контроль	Залік
Викладач	ст. викладач Алещенко О.В.

Технології	Технології розроблення серверного програмного забезпечення (Back-end)
Кафедра	Обчислювальної техніки
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 100
Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: Програмування, Об'єктно-орієнтоване програмування, Алгоритми та методи обчислень, Системне програмування, Структури даних і алгоритми, Дискретна математика, Інженерія програмного забезпечення.
Що буде вивчатися	Дисципліна має на меті: надання ґрунтовних теоретичних знань і конкретних практичних навичок розробки серверного програмного забезпечення. Навчальна дисципліна спрямована на формування компетенцій з розробки Back-end частини прикладних Web-застосунків. В поєднанні із навчальною дисципліною «Технології програмування користувацьких інтерфейсів (Front-End)» (8 семестр) забезпечується набуття комплексних знань і практичних навичок проектування і розробки прикладного програмного забезпечення, що базується на сучасних і перспективних клієнт-серверних технологіях розподіленої обробки і обміну інформацією. Теоретична і практична складові навчальної дисципліни спрямовані на набуття компетенцій у ході реалізації конкретного проекту з розробки Back-end частини із спрощеною реалізацією Front-end складової. Дисципліна складається із наступних блоків: 1. Методологія і технології побудови та створення клієнт-серверних (Web) застосунків (модель “клієнт-сервер”, хмарні сервери, рівень мереж, протокольний обмін та інтернет стандарти: TCP/IP, WWW, XML, JSON, HTTP, HTTPS, FTP, Telnet, MIME; URL адреси та

	<p>ідентифікатори URI; основні засоби дослідження помилок (ICMP, ping, traceroute); сокети, IP і адресація портів; використання проксі-сервера; сервіси пошуку імен: DNS, whois; сервіси віддаленого доступу: Telnet, SSH, Remote Desktop, VNC);</p> <p>2. Поняття API та його протоколи. Організація системи (правил) обмеження доступу та авторизації. Фреймворки та набори інструментів для розробки Back-end. Взаємодія з БД шляхом використання менеджера об'єктів (ORM - Hibernate, Doctrine, typeORM, Eloquent). Технології розробки серверного (Back-end) програмного забезпечення: PHP (Symfony, Laravel), Java (Spring Boot, Java EE), JavaScript (Express, NestJS);</p> <p>3. Специфікація веб сервісів. Тестування коду (необхідність розроблення специфікації; OPENAPI3 - документація; створення схем (яких?); основи Test Driven Development - unit, integration та e2e тести (PHPUnit, JUnit, Jest); перехоплення та оброблення помилок.</p> <p>4. Вступ в DevOps: складові частини DevOps – Development (Software Engineering), Operations, Quality; Assurance; налаштування серверів Apache, PHP, MySQL; інтеграція DevOps в процес розробки програмного забезпечення; DevOps and Digitalization; CI / CD development; Забезпечення високої доступності сервісів – High Availability.</p>
<b>Чому це цікаво/треба вивчати</b>	Опанування знань, умінь та навичок, що потребують посади: Back-end developer; Back-end Engineer; Back-end Team Lead; DevOps Engineer тощо.
<b>Чому можна навчитися (результати навчання)</b>	<p>1. Знання та навички застосування і реалізації клієнт-серверних технологій для створення ефективних Back-end частин сучасних Web застосунків: методологія і технології побудови та створення Back-end частин клієнт-серверних застосунків; сучасні технології розробки серверного програмного забезпечення; тестування коду; основи реалізації DevOps завдань.</p> <p>2. Знання і навички застосування базових Back-end – підходів: фреймворки, тестування, розробка API.</p>
<b>Як можна користуватися набутими знаннями і уміньми (компетентності)</b>	<p>Набуті компетенції з розроблення серверного програмного забезпечення (Back-end) можливо застосовувати на проектах наступних прикладних галузей і технологій:</p> <ol style="list-style-type: none"> <li>1. Інтернет-комерція;</li> <li>2. Електронний документообіг;</li> <li>3. CRM-системи різного прикладного спрямування, побудованих на технологіях розподілених процесах збору, зберігання, обробки і обміну інформацією.</li> </ol>
<b>Інформаційне забезпечення</b>	Силабус дисципліни, навчально-методичний комплекс.
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	Асистент Валько Володимир Володимирович

Дисципліна	Розробка системних програм
Кафедра	Обчислювальної техніки
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 100
Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: Програмування, Об'єктно-орієнтоване програмування, Алгоритми та методи обчислень, Системне програмування, Алгоритми і структури даних, Вступ до операційної системи Linux
Що буде вивчатися	<ul style="list-style-type: none"> <li>— склад системного програмного забезпечення сучасних комп'ютерних систем;</li> <li>— технологія покрокової розробки складних програмних продуктів Incremental Approach на прикладі розробки системних програм;</li> <li>— практичне застосування цієї технології під час розробки системних програм;</li> <li>— тестування розроблених системних програм.</li> </ul>
Чому це цікаво/треба вивчати	<ul style="list-style-type: none"> <li>— системне програмне забезпечення присутнє у будь-якому обчислювальному пристрої, тому попит на тих, хто розуміється у ньому - завжди буде високим;</li> <li>— системне програмне забезпечення є основою функціонування комп'ютерних систем, саме воно диктує вимоги до прикладних програм, тому його розробники будуть завжди попереду програмістів - прикладників;</li> <li>— розробкою системного програмного забезпечення займаються, як правило, найбільш відомі компанії, які визначають тренд розвитку комп'ютерних технологій на багато років вперед.</li> </ul>
Чому можна навчитися (результати навчання)	— розробляти та тестувати системні програми, розуміти повідомлення операційних систем, виконувати оптимізацію коду.
Як можна користуватися набутими знаннями і уміннями (компетентності)	— при розробці та удосконаленні операційних систем, спеціалізованих програмних систем, систем комп'ютерної безпеки, систем програмування, окремих складових системного ПЗ: драйверів, програм керування файлами тощо.
Інформаційне забезпечення	Силабус дисципліни, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи, самостійна робота.
Семестровий контроль	Залік
Викладач	к.т.н., доцент Павлов В. Г.

Дисципліна	Математичні основи захисту даних та інформаційної безпеки
Кафедра	Обчислювальної техніки
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	4 курс, 7 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з архітектури комп'ютера, комп'ютерних систем та комп'ютерних мереж, алгоритмів та структур даних, дискретної математики, теорії ймовірності та математичної статистики, системного програмування.
Що буде вивчатися	Математичні основи сучасної криптографії: незворотні перетворення булевої алгебри, теорії чисел, еліптичних кривих та кінцевих полів Галуа. Методи синтезу криптографічних булевих перетворень, побудова на їх основі алгоритмів захисту даних. Сучасні алгоритми та протоколи захисту інформації на основі булевих функцій. Атаки на алгоритми та протоколи цього класу. Методи побудови систем криптографічного захисту на основі незворотних перетворень теорії чисел. Алгоритми та протоколи захисту інформації на основі незворотних перетворень теорії чисел та методи їх зламу. Математичні основи організації гомоморфного шифрування даних при їх обробці в хмарах. Методи зламу засобів захисту даних шляхом часового чи статичного аналізу динаміки споживання потужності. Математичні засади програмного поліморфізму при захищеній організації обчислень. Квантові обчислення та квантова криптографія.
Чому це цікаво/треба вивчати	Поступальний розвиток інформаційної інтеграції вимагатиме якісно нових засобів захисту даних та розмежування доступу до них в інтегрованих середовищах. Створення та використання таких засобів неможливе без оволодіння фундаментальними основами криптографії. Професійний аналіз реального рівня захищеності даних та програм вимагає наявності спеціальної математичної підготовки.
Чому можна навчитися (результати навчання)	Отримати ґрунтовні знання в спеціальних розділах математики, які лежать в основі сучасних та перспективних засобах криптографічного захисту даних. Оволодіти навичками створення базових блоків криптографічного захисту даних, побудови на їх основі алгоритмів та протоколів. Набути вмінь аналізувати рівень захищеності засобів інформаційної безпеки. Знати протоколи криптографічного захисту інформації в комп'ютерних системах та мережах.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розробляти, вдосконалювати та адаптувати під умови конкретного використання програмні і апаратні засоби криптографічного захисту даних. Створювати гомоморфні шифри для захисту даних безпосередньо в процесі їх віддаленої обробки в хмарах залежно від її характеристик. Здійснювати професійний аналіз інформаційної безпеки даних з огляду на здатність протистояння різним видам атак.
Інформаційне забезпечення	Силабус дисципліни, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи, самостійна робота



<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., доцент Марковський О. П.

### 3 курс 6 семестр

#### Освітній компонент 5Ф-каталогу

Дисципліна	Аналіз даних з використанням мови Python
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 5 семестр
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: програмування, алгоритми та структури даних, математичний аналіз, дискретна математика, теорія ймовірностей.
Що буде вивчатися	<ul style="list-style-type: none"> <li>- основні бібліотеки мови Python для роботи з даними (NumPy, SciPy, Matplotlib, Seaborn, Scikit-learn, Pandas) та їх використання;</li> <li>- статистична обробка даних в SciPy та Pandas;</li> <li>- попередня підготовка даних для аналізу в Pandas та Scikit-learn;</li> <li>- інструменти Python для візуалізації даних;</li> <li>- основи машинного навчання та використання його базових алгоритмів за допомогою бібліотеки Scikit-learn.</li> </ul>
Чому це цікаво/треба вивчати	Кожна людина в своїй професійній діяльності має справу зі значними обсягами даних. Аналіз цих даних дозволяє правильно їх інтерпретувати, отримати нову інформацію, допомагає приймати рішення. Python є популярною мовою програмування, яка активно використовується та має широкий інструментарій для аналізу даних.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> <li>- вдосконалити навички роботи з Python;</li> <li>- ефективно використовувати бібліотеки мови Python для обробки, візуалізації та аналізу даних;</li> <li>- правильно обирати методи та алгоритми для попередньої обробки даних;</li> <li>- правильно обирати найбільш інформативні способи представлення даних;</li> <li>- використовувати алгоритми машинного навчання для класифікації, кластеризації, регресійного аналізу і т.д.</li> </ul>
Як можна користуватися набутими знаннями і вміннями (компетентності)	Ефективно обробляти, візуалізувати та аналізувати дані, отримані внаслідок власних експериментів та досліджень, створювати програми мовою Python для роботи з даними.
Інформаційне забезпечення	Силабус дисципліни, лекції, лабораторні роботи
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	ст. викладач Тимофєєва Ю.С.

Дисципліна	Розробка мобільних застосунків під Android
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 150
Курс, семестр	3 курс, 6 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з теорії алгоритмів та структур даних, об'єктно-орієнтованого програмування, основ веб-програмування, роботи з базами даних та знання мов програмування Java та SQL.
Що буде вивчатися	<p>Програма курсу передбачає:</p> <ul style="list-style-type: none"> <li>• вивчення архітектури платформи Android</li> <li>• розробку нативних додатків на мові Java</li> <li>• роботу з файловою системою та базами даних SQLite</li> <li>• доступ до глобальної мережі та розробки веб-додатку</li> </ul> <p>опанування роботи з датчиками та мультимедіа</p>
Чому це цікаво/треба вивчати	Даний курс орієнтований на опанування розробки мобільних додатків під Android як людьми з мінімальним рівнем знання програмування так і людьми, які хочуть покращити свої знання з певних нюансів. Після його завершення ви зможете розробляти мобільні застосунки під платформу Android як нативні, так і веб-застосунки під ваші завдання.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> <li>• пізнати основи розробки дизайну та функціоналу додатків під платформу Android</li> <li>• розробляти нативні додатки;</li> <li>• розробляти гібридні додатки;</li> <li>• організувати сховища для збереження даних у пам'яті пристрою або у базі даних;</li> <li>• використовувати інтернет-джерела для доступу до мультимедійних даних;</li> </ul> <p>оперувати датчиками для роботи з застосунками</p>
Як можна користуватися набутими знаннями і вміннями (компетентності)	<ul style="list-style-type: none"> <li>• аналізувати та обирати тип мобільного додатку;</li> <li>• розробляти мобільні версії сайтів;</li> <li>• застосовувати різні інтерактивні елементи для зручного інтерфейсу мобільного застосунку;</li> </ul> <p>аналізувати та обирати сховища для збереження даних з урахуванням безпеки по доступу</p>
Інформаційне забезпечення	Силабус дисципліни, навчально-методичний комплекс.
Форма проведення занять	Лекції та лабораторні роботи
Семестровий контроль	Залік
Викладач	PhD, асистент Орленко С.П.

Дисципліна	Основи Back-end технологій
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 150
Курс, семестр	3 курс, 6 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: Базові знання мов програмування, основ графіки та дизайну, роботи в мережі інтернет.
Що буде вивчатися	Розробка серверних додатків мовою <b>Node.js</b> . Робота з мережними запитами. Аутентифікація та авторизація. REST API, WebSockets.Socket.IO. Розробка серверних додатків мовою <b>PHP</b> . Робота з базою даних MySQL. Підключення веб-додатків до бази даних з метою зберігання та обміну інформацією між базою даних і додатком.
Чому це цікаво/треба вивчати	Back-end developer — це спеціаліст, який займається програмно-адміністративною частиною веб-додатка, внутрішнім змістом системи, серверними технологіями – базою даних, архітектурою, програмною логікою.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> <li>▪ розробляти серверні додатки мовою Node.js;</li> <li>▪ розробляти серверні додатки мовою PHP;</li> </ul> розробляти структуру БД та забезпечувати підключення веб-додатків до бази даних;
Як можна користуватися набутими знаннями і уміннями (компетентності)	При проектуванні серверної частини сайту, розробки движка сайту та серверних додатків, необхідних для роботи сайту. При проектуванні структури бази даних сайту та для організації взаємодії між базою даних та WEB-додатком.
Інформаційне забезпечення	Силабус дисципліни, навчально-методичний комплекс, VSC, GitHUB, Node.js, PHP, MySQL
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	к.т.н., доцент Голубев Л.П.

Дисципліна	Мультипарадигмне програмування
Кафедра	Інформатики та програмної інженерії
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 6 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	<ul style="list-style-type: none"> <li>• Знання мінімум однієї мови програмування та досвід створення та відлагодження програм. Основи дискретної математики.</li> </ul>

<b>Що буде вивчатися</b>	<ul style="list-style-type: none"> <li>• Основи застосування методів різних парадигм програмування при проектуванні додатків</li> <li>• Принципи використання функціонального програмування</li> <li>• Основи функціонального програмування в Scala. Створення функцій вищого порядку в Scala</li> <li>• Основи функціонального програмування в Clojure . Паралельне програмування в Clojure</li> <li>• Основи функціонального програмування в Haskell. Узгодження шаблонів Функції рекурсії, контролю та вищого порядку в Haskell</li> <li>• Основи логічного програмування в Prolog. Штучний інтелект та обчислювальна лінгвістика в Prolog</li> </ul>
<b>Чому це цікаво/треба вивчати</b>	<ul style="list-style-type: none"> <li>• Логічне програмування є найбільш корисним у таких сферах, які стосуються досліджень штучного інтелекту, таких як вирішення проблем, (планування) шляху чи тлумачення природних мов. Деякі сфери, які не є строго AI, такі як парсер, також можуть значною мірою отримати користь від prolog.</li> <li>• Однак ті, хто не працює з природними мовними системами щодня, також можуть отримати велику користь від Prolog. Оскільки Prolog використовує певні методи програмування, які вважаються складними або вдосконаленими (особливо це стосується рекурсії), Prolog може бути чудовим способом отримання розуміння цих методів. Практикуючий програміст Prolog, безумовно, буде глибоко розуміти такі поняття, як рекурсія, пошук проблемних дерев та логічне програмування обмежень. Prolog навіть може слугувати інструментом "замальовки", щоб спробувати основні ідеї програмування, перш ніж реалізувати їх на процедурних мовах. Через тісні зв'язки з логікою предикатів, вивчення Prolog може призвести до більш чіткого розуміння логіки предикатів.</li> </ul>
<b>Чому можна навчитися (результати навчання)</b>	аналізувати, аргументувати, приймати рішення при розв'язанні спеціалізованих задач та практичних проблем з використанням підходи із функціональної та логічної парадигми
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>• Створювати функціональні програми за допомогою Haskell Scala та Clojure</li> <li>• Створювати логічні програми за допомогою Prolog</li> </ul> Використовувати функціональні і логічні підходи в ООП
<b>Інформаційне забезпечення</b>	Силабус дисципліни, контрольні завдання, тексти лекцій
<b>Форма проведення занять</b>	Лекції та лабораторні (комп'ютерні практикуми)
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	доц., к.т.н. Баклан І.В.

<b>Дисципліна</b>	<b>Сучасні технології розробки та управління життєвим циклом ПЗ</b>
<b>Кафедра</b>	Інформатики та програмної інженерії
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 30 до 60

<b>Курс, семестр</b>	3 курс, 6 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Основні знання з програмування, алгоритмів та структур даних, основи роботи з системами контролю версій.
<b>Що буде вивчатися</b>	Методології розробки ПО, управління версіями, CI/CD, тестування ПО, контейнеризація та оркестрація, облачні технології, моніторинг і логування, безпека ПО, архітектура ПО, управління проектами в IT.
<b>Чому це цікаво/треба вивчати</b>	Сучасний ринок праці висуває високі вимоги до якості та ефективності процесів розробки ПО. Вивчення цих технологій дозволить студентам бути конкурентоспроможними в IT-індустрії, розуміти та застосовувати найкращі практики та інструменти.
<b>Чому можна навчитися (результати навчання)</b>	Розробка високоякісного ПО з використанням сучасних методологій і інструментів, автоматизація процесів розробки та впровадження, забезпечення безпеки та високої доступності програмних продуктів.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Розробка та підтримка ПО в компаніях різного масштабу, впровадження інноваційних технологій для підвищення ефективності розробки та експлуатації ПО, управління командами розробників та проектами.
<b>Інформаційне забезпечення</b>	Навчальні матеріали, лекції, лабораторні роботи, доступ до хмарних платформ, література та онлайн-ресурси з сучасних технологій розробки ПО.
<b>Форма проведення занять</b>	Лекції, лабораторні роботи.
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	доц., к.т.н. Губський А. М.

<b>Дисципліна</b>	<b>Технології програмування на ПЛІС (FPGA)(Сертифікатна програма)</b>
<b>Кафедра</b>	Обчислювальної техніки
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 90
<b>Курс, семестр</b>	3 курс, 6 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з комп'ютерної логіки, комп'ютерної арифметики, дискретної математики. Базові знання архітектури комп'ютера, що включають знання загальної конструктивно-функціональної організації комп'ютерів архітектури Фон-Неймана, загальні принципи мікропрограмного керування, структуру процесорного ядра та загальні принципи організації мікропрограмного рівня. Навички програмування.
<b>Що буде вивчатися</b>	Методи цифрового дизайну та проектування для FPGA. Мови опису апаратури для розроблення цифрових пристроїв. Проектування цифрових пристроїв з використанням Verilog. Функціональне моделювання та структурний синтез в сучасних САПР.

	<p>Практичні завдання побудовані спочатку на простих прикладах програмування вузлів комп'ютерної техніки. Далі пропонується знайомство з основними етапами повного циклу розроблення IP-ядер цифрової техніки, включаючи функціональний синтез, структурний синтез, верифікацію розробок та програмування мікросхеми ПЛІС. Розглядаються базові питання створення TestBench рівнів на мові Verilog. Для перевірки правильності створених проектів та налагодження використовуються САПР ModelSym MentorGraphic, Quartus II. Розглядаються питання інтеграція САПР Quartus II и ModelSim. Для експериментів використовується налагоджувальна плати компанії Altera DE2 Board (Cyclon II), DE10 Standard Board (Cyclon IV), DE1 SoC Board (Cyclon V).</p>
<p><b>Чому це цікаво/треба вивчати</b></p>	<p>Програмована логіка (FPGA) останнім часом стає основною технологією, що використовується для створення електронних систем в різних галузях застосування. Отримані навички та знання будуть актуальні в областях розроблення рішень для Інтернету Речей (IoT), вбудованих систем, розумних систем. Розроблення апаратних систем для автомобільної промисловості, промислової автоматизації, високопродуктивних обчислень, штучного інтелекту машинного навчання.</p> <p>Курс «Технології програмування на ПЛІС (FPGA)» може бути прослуханий студентами як окремий курс, так і у складі <b>сертифікатної програми «Інженерія вбудованих систем та Інтернет речей»</b>, яка запроваджена на факультеті інформатики та обчислювальної техніки з 2021 – 2022 навчального року. Детально з сертифікатною програмою можна ознайомитися на сайті кафедри обчислювальної техніки <a href="https://comsys.kpi.ua/sertifikatni-programi">https://comsys.kpi.ua/sertifikatni-programi</a>.</p> <p>Перелік дисциплін сертифікатної програми:</p> <ul style="list-style-type: none"> <li>• Технології програмування C/Embedded – 5 семестр;</li> <li>• Управління IT-інфраструктурними проектами – 6 семестр;</li> <li>• Технології програмування на ПЛІС (FPGA) – 6 семестр;</li> <li>• Тестування та контроль якості (QA) вбудованих систем – 7 семестр;</li> <li>• Технології розроблення вбудованих систем IoT – 8 семестр.</li> </ul>
<p><b>Чому можна навчитися (результати навчання)</b></p>	<p>Проектувати цифрові схеми та пристрої з використанням сучасних методів проектування та мов проектування апаратури. Отримати навички програмування на мові Verilog. Користуватися сучасними професійними засобами для цифрового дизайну та моделювання. Виконувати функціональне моделювання і налагодження пристроїв в САПР ModelSym. Виконувати структурний синтез в САПР Quartus II Altera. Реалізувати пристрої на сучасних платах компанії Altera DE2 Board, DE10 Board, DE1 SoC Board.</p>
<p><b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b></p>	<p>Базові знання у використанні мови проектування апаратури, Verilog, а також досвід і навички використання сучасних САПР і продуктів компанії Altera достатньо для продовження професійного навчання в напрямку проектування апаратури для складних систем на FPGA та ASIC, систем-на-кристалі (SoC), вбудованих систем, розумних систем. програмно-апаратних реалізацій систем штучного інтелекту. Отримані базові знання програмування та методів цифрового дизайну дозволять за необхідності швидко перейти на використання мови програмування апаратури VHDL, мікросхем інших виробників FPGA (Xilinx), а також інших САПР для функціонального та структурного</p>

	синтезу на FPGA та ASIC (Synopsys, Aldec, Cadence). Отримані базові знання допоможуть у подальшому професійному навчанні в тому числі самостійно.
<b>Інформаційне забезпечення</b>	Силабус дисципліни, навчально-методичний комплекс.
<b>Форма проведення занять</b>	Лекції, практичні та семінарські заняття, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	д.т.н., професор Клименко І.А., асистент Гайдай А.Р.

<b>Дисципліна</b>	<b>Системне мислення</b>
<b>Кафедра</b>	Інформаційних систем та технологій
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 25 до 120
<b>Курс, семестр</b>	3 курс, 6 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін «Філософія», «Вища математика», «Комп'ютерні мережі», «Інтернет речей», «Дискретна математика», «Системна інженерія».
<b>Що буде вивчатися</b>	Шляхи вирішення комплексних проблем життєдіяльності складних інженерно-технічних систем, а також практичний інструментарій синтезу таких систем та підтримки їх руху за стадіями життєвого циклу.
<b>Чому це цікаво/треба вивчати</b>	Світ, який нас оточує — складна система. Ми не розуміємо багатьох речей, а тому мимохіть стаємо їхніми заручниками. Чим більше зусиль ми докладаємо, то менш досяжною видається наша мета, а старі проблеми повертаються й накопичуються і погіршують і без того поганий стан речей. Щоб узяти контроль над власним життям, варто навчитися системного мислення, адже тільки розуміючи справжнє підґрунтя подій, ми можемо змінювати на краще власні дії та довколишній світ. Системне мислення впорядковує думки та вчить бачити закономірності в подіях. Це вміння дає нам змогу підготуватися і навіть впливати на ці події.
<b>Чому можна навчитися (результати навчання)</b>	Здобувачі повинні досягти таких програмних результатів навчання: - мати передові концептуальні та методологічні знання з ІСТ і намежі предметних галузей, а також дослідницькі навички, достатні для проведення наукових і прикладних досліджень на рівні останніх світових досягнень з відповідного напрямку, отримання нових знань та/або здійснення інновацій; - застосовувати сучасні інструменти і технології пошуку, оброблення та аналізу інформації, зокрема, статистичні методи аналізу даних великого обсягу та/або складної структури, спеціалізовані бази БД та ІС; - планувати і виконувати експериментальні та/або теоретичні дослідження з використанням сучасних інструментів, критично



	аналізувати результати власних досліджень і результати інших дослідників у контексті усього комплексу сучасних знань щодо досліджуваної проблеми; - вміти розв'язувати складні системні та спеціалізовані питання при розробці ІС; застосовувати методи математичного моделювання та оптимізації при реалізації науково-дослідних проєктів; - знати загальні тенденції розвитку новітніх ІТ у передових країнах, оцінювати ефективність передових технологій, впроваджувати найбільш ефективні методи та прийоми для розроблення сучасних технологій.
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	Знання та вміння забезпечують та надають можливість фахівцю не тільки впевнено вирішувати технічні та наукові задачі але й з великою точністю прогнозувати складні інженерно-технічні системи.
<b>Інформаційне забезпечення</b>	Силабус дисципліни, опорний конспект лекцій, методичні вказівки до виконання лабораторних робіт, перелік завдань для самостійної роботи студента, перелік питань до МКР та семестрового контролю.
<b>Форма проведення занять</b>	Лекції, лабораторні роботи.
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., доцент Сокульський О.Є.

<b>Дисципліна</b>	<b>Методи чисельної оптимізації</b>
<b>Кафедра</b>	Обчислювальної техніки
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс</b>	3, семестр 6
<b>Обсяг</b>	4 кредити ЄКТС
<b>Мова викладання</b>	Українська
<b>Кафедра</b>	Обчислювальної техніки
<b>Вимоги до початку вивчення</b>	Для вивчення освітнього компоненту «Методи чисельної оптимізації» потрібно володіти компетентностями з наступних дисциплін: «Вища математика», «Аналітична геометрія та лінійна алгебра», «Дискретна математика» та «Програмування».
<b>Що буде вивчатися</b>	Вивчатимуться методи чисельної оптимізації, які використовуються на практиці у машинному навчанні.
<b>Чому це цікаво/треба вивчати</b>	Навчання сучасних архітектур глибоких нейронних мереж, яке здійснюється на великих наборах даних, може займати години, дні або навіть тижні. Продуктивність алгоритму оптимізації безпосередньо впливає на ефективність навчання моделі. Саме тому нещодавно було розроблено велику кількість нових методів для оптимізації навчальних параметрів нейронних мереж і зменшення часу навчання моделей. Розуміння принципів алгоритмів оптимізації та роль їх гіперпараметрів дозволить студентам цілеспрямовано налаштувати оптимізаційний алгоритм та адаптувати його до поставленої задачі з метою покращення продуктивності моделей машинного навчання.

<b>Чому можна навчитися (результати навчання)</b>	Отримання досвіду використання та налаштування оптимізаційних алгоритмів для вирішення прикладних задач машинного навчання.
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	Набуті знання можна використовувати під час вирішення прикладних задач машинного навчання.
<b>Інформаційне забезпечення</b>	Силабус, навчальні посібники, завдання на лабораторні роботи
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	Асистент Кочура Юрій Петрович

## Освітній компонент БФ-каталогу

Дисципліна	Штучний інтелект в задачах обробки зображень
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 150
Курс, семестр	3 курс, 6 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з: математичний аналіз, лінійна алгебра, будь-якої мови програмування (бажано базові знання з Python), розуміння ООП.
Що буде вивчатися	При проходженні даної дисципліни, студенти познайомляться з поняттям штучного інтелекту та його розділом машинним навчанням, з термінологією та основними поняттями з обробки та аналізу цифрових зображень, з сучасними технологіями візуального покращення та відновлення зображень. Познайомляться з алгоритмами покращення та відновлення на півтонових та кольорових зображень; з морфологічною обробкою зображень. На лабораторних заняттях студенти будуть обробляти та аналізувати зображення засобами бібліотеки PythonOpenCV, виконувати перетворення зображень у різні формати, добувати корисну інформацію з зображень; використовувати традиційні технології розпізнавання образів на наборах даних (наприклад CIFAR-10). Проектувати та навчати нейронні мережі в середовищі Keras та TensorFlow для класифікації зображень
Чому це цікаво/треба вивчати	<ol style="list-style-type: none"> <li>1. Вам подобається фотографувати і Вам було б цікаво навчитись обробляти та аналізувати зображення за допомогою Python?</li> <li>2. Вам цікаво зрозуміти як відбувається розпізнавання людей, обличчя або номерів на авто з відеокамер?</li> <li>3. Вас цікавлять математичні основи вбудованих бібліотек Python для розпізнавання зображень?</li> <li>4. Було б цікаво навчити нейронну мережу розпізнавати рукописний текст?</li> </ol> <p>Якщо хоча б на одне з питань Ви відповіли – ТАК, то даний курс саме для Вас</p>
Чому можна навчитися (результати навчання)	Сучасними методам машинного навчання та вмінню застосовувати сучасні технології та класичні алгоритми при розробляти програмне забезпечення для обробки та аналізу зображень.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Отримані знання та уміння в даному курсі можуть стати підґрунтям для обрання теми бакалаврської або магістерської роботи, для подальших наукових досліджень. Уміння розробляти програмне забезпечення для обробки та аналізу зображень можна застосувати при розробці комп'ютерних ігор, в напрямку «Розумний будинок»
Інформаційне забезпечення	Силабус дисципліни, конспект лекцій (навчальний посібник), перелік питань до екзамену, перелік завдань до екзамену, навчальний посібник до виконання лабораторних робіт, завдання до виконання лабораторних робіт, перелік питань і завдань для проведення поточного і підсумкового контролю, завдання для комплексної контрольної роботи, завдання до самостійної роботи студентів та

	методичні вказівки до їх виконання.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	д.т.н., професор Онищенко В.В.

Дисципліна	Основи візуального програмування
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 6 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з <i>вищої математики, теорії алгоритмів, класичних мов та технологій програмування</i>
Що буде вивчатися	Підходи, засоби та технології візуального програмування виконавчих платформ в задачах автоматизації інженерних досліджень. Основні можливості програмного комплексу NI LabVIEW
Чому це цікаво/треба вивчати	Альтернативний підхід до розробки програмних рішень із використанням технологій візуального програмування дозволяє надшвидкими темпами отримати готові рішення зокрема систем моніторингу в інформаційних системах, збору та обробки даних з мінімальним часом відлагодження та виправлення помилок.
Чому можна навчитися (результати навчання)	Базові навички візуального програмування апаратних платформ автоматизованих інформаційних систем керування збору та обробки даних засобами LabVIEW.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розробляти програмні рішення в сфері автоматизації інженерних досліджень, розумний дім та автоматизованих інформаційних системах керування технологічними/виробничими процесами
Інформаційне забезпечення	Силабус, навчально-методичний комплекс
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	ст. викладач Яланецький В.А.

Дисципліна	WEB – орієнтовані технології. Основи Frontend та Backend розробок
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 6 семестр
Обсяг	4 кредити, 120 годин

<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: «Програмування 1. Основи програмування», «Програмування - 2. Структури даних та алгоритми», «Теорія алгоритмів», «Комп'ютерні мережі», «Бази даних», «Основи клієнтської розробки», «Розроблення веб-застосувань».
<b>Що буде вивчатися</b>	<b>Дана навчальна дисципліна</b> є основою теоретичних знань, практичних вмінь та навичок розробки WEB-ресурсів та WEB-орієнтованих застосунків, які складаються з Frontend та Backend розробок. Це є основою у підготовці фахівців з інформаційних технологій для проектування, розробки, впровадження WEB - орієнтованих технологій у професійній діяльності.
<b>Чому це цікаво/треба вивчати</b>	<b>Метою</b> дисципліни «WEB-технології. Frontend та Backend розробок» є засвоєння високого рівня необхідних знань та умінь з основ WEB – орієнтованих технологій, формування у студентів здатності проектувати, розробляти WEB-сайти та WEB-орієнтовані застосунки (Frontend та Backend розробки)
<b>Чому можна навчитися (результати навчання)</b>	<p><b>За результатами вивчення дисципліни студент повинен:</b></p> <p><b>Знати:</b></p> <ul style="list-style-type: none"> <li>— структуру HTML- документу;</li> <li>— каскадні таблиці стилів (CSS), основні властивості CSS;</li> <li>— основні компоненти об'єктної моделі документа (DOM);</li> <li>— правила використання і основні елементи мов JavaScript та PHP.</li> <li>— правила використання платформ та фреймворків для Frontend та Backend розробок</li> </ul> <p><b>Вміти:</b></p> <ul style="list-style-type: none"> <li>— розробити структуру WEB- сайту;</li> <li>— застосовувати засоби каскадних таблиць стилів (CSS) для форматування сторінок WEB-сайту;</li> <li>— застосовувати засоби мови JavaScript при роботі з основними об'єктами HTML-документу;</li> <li>— застосовувати сучасні фрейворки для написання Frontend частини WEB-застосунку;</li> <li>— розробити базу даних для використання у динамічних WEB-сайтах;</li> <li>— використовувати платформу Node.js для написання Backend частини WEB-застосунку та зв'язку з базами даних;</li> <li>— використовувати засоби мови PHP для написання серверних скриптів та зв'язку з базами даних;</li> <li>— бути здатним вирішити складні спеціалізовані задачі та практичні проблеми у певній галузі професійної діяльності або навчання.</li> </ul> <p>Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання інформаційних систем та технологій.</p> <p>Демонструвати знання сучасного рівня технологій інформаційних систем, практичні навички програмування та використання прикладних і спеціалізованих комп'ютерних систем та</p>

	<p>середовищ з метою їх запровадження у професійній діяльності. Демонструвати вміння аналізувати вимоги та розробляти веб-додатки, веб-сервіси, веб-сайти з використанням базових принципів, сучасних технологій та мов програмування для створення веб-застосувань, як інтерфейсу доступу до сервісів ІТ-інфраструктури</p>
<p>Як можна користуватися набутими знаннями і уміннями (компетентності)</p>	<p><b>Програмні компетентності:</b></p> <ul style="list-style-type: none"> <li>— Здатність розв’язувати складні спеціалізовані задачі та практичні проблеми в області інформаційних систем та технологій, або в процесі навчання, що характеризуються комплексністю та невизначеністю умов, які потребують застосування теорій та методів інформаційних технологій.</li> </ul> <p><b>Загальні компетентності:</b></p> <ul style="list-style-type: none"> <li>— Здатність до абстрактного мислення, аналізу та синтезу;</li> <li>— Здатність застосовувати знання у практичних ситуаціях;</li> <li>— Здатність до розуміння предметної області та професійної діяльності;</li> <li>— Здатність вчитися і оволодівати сучасними знаннями;</li> <li>— Здатність до пошуку, оброблення та узагальнення інформації з різних джерел;</li> <li>— Здатність розробляти та управляти проектами;</li> <li>— Здатність оцінювати та забезпечувати якість виконуваних робіт;</li> </ul> <p><b>Спеціальні (фахові, предметні) компетентності:</b></p> <ul style="list-style-type: none"> <li>— Здатність застосовувати технології та інструментальні засоби для розроблення веб-застосувань, веб-сервісів, веб-сайтів та веб-інтерфейсів з інтеграцією зовнішніх даних і програмних продуктів;</li> <li>— Здатність вирішувати задачі інтеграційних процесів інформаційних систем у сфері виробництва та керування з використанням методів аналізу та синтезу засобів передачі, зберігання та обробки інформації, основ сервіс-орієнтованого підходу до обслуговування користувачів інформаційних систем, базових та прикладних інформаційних технологій та інструментальних засобів інфраструктури ІТ;</li> <li>— Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші) ;</li> <li>— Здатність аналізувати об’єкт проектування або функціонування та його предметну область;</li> <li>— Здатність управляти якістю продуктів і сервісів інформаційних систем та технологій протягом їх життєвого циклу</li> <li>— Здатність формувати нові конкурентоспроможні ідеї й реалізовувати їх у проектах (стартапах).</li> </ul> <p>Здатність управляти та користуватися сучасними інформаційно-комунікаційними системами та технологіями (у тому числі такими, що базуються на використанні Інтернет).</p>
<p>Інформаційне забезпечення</p>	<p>Силабус дисципліни, конспект лекцій, методичні вказівки до виконання лабораторних робіт, список основної та допоміжної літератури.</p>
<p>Форма проведення занять</p>	<p>Лекції, лабораторні роботи, робота в командах</p>

Семестровий контроль	Залік
Викладач	ст. викладач Проскура С.Л.

Дисципліна	Робота з даними в хмарних середовищах
Кафедра	Інформатики та програмної інженерії
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 6 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання мов програмування та роботи з комп'ютером, англійська мова на рівні pre-intermediate. Бажання набути знань та навичок в роботі з даними з точки зору розробника та адміністратора.
Що буде вивчатися	<ul style="list-style-type: none"> <li>• Теорія даних, робота та функції СУБД (систем управління базами даних), архітектура баз даних</li> <li>• SQL (structured query language)</li> <li>• Технології резервного копіювання та відновлення даних</li> <li>• Побудова звітності та візуалізація даних</li> <li>• Засоби забезпечення високої доступності</li> <li>• Хмарні середовища AWS, Azure, Google Cloud</li> </ul> Захист даних
Чому це цікаво/треба вивчати	Збереження та обробка даних – це одні з перших задач, які виникли перед ІТ-індустрією. Будь-який спеціаліст в галузі, включаючи тестувальників, розробників, адміністраторів та архітекторів, в тій-чи іншій мірі повинні знати та вміти дістати потрібну інформацію, подати її в потрібному вигляді, впевнитись в тому, що вона не спотворена і інші користувачі та системи можуть швидко та безпечно зробити те ж саме. З впровадженням хмарних технологій, що дозволяють мінімізувати час розгортання СУБД та зробити дані доступними в будь-який час та з будь-якої точки земної кулі, подібні навички стали тільки більш актуальними.
Чому можна навчитися (результати навчання)	Буде вивчено повний цикл від проектування БД та обрання хмарного рішення для її розміщення до забезпечення високої доступності та захисту інформації.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розуміння як можна отримати потрібні дані та чи є вони вразливими. Мігрувати дані, створювати репліки, резервні копії, тощо. Писати запити та оптимізувати їх для швидкої маніпуляції даними. Проектувати бази даних з нуля для будь-яких проектів.
Інформаційне забезпечення	Силабус дисципліни, контрольні завдання, навчальні матеріали.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	ас.Ахаладзе А.Е.

Дисципліна	Сучасні операційні системи
Кафедра	Інформатики та програмної інженерії
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 6 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Знання основ комп'ютерних систем і мереж, алгоритмів та структур даних, підходів до програмування, засобів візуалізації інтерфейсу користувача.
Що буде вивчатися	<ul style="list-style-type: none"> <li>• Особливості сучасних операційних систем.</li> <li>• Процес організації роботи операційних систем.</li> <li>• Підходи до архітектури операційних систем.</li> <li>• Практики використання можливостей операційних систем.</li> </ul> Процес пошуку проблем швидкодії та практики оптимізації.
Чому це цікаво/треба вивчати	Внаслідок прискороного переходу більшості користувачів на мобільні комп'ютерні пристрої та все чіткішої сегментації способів використання комп'ютерної техніки в різних сферах діяльності людини зростає попит на знання актуальних операційних систем від всіх учасників ІТ-індустрії.
Чому можна навчитися (результати навчання)	Сформувати уявлення про внутрішність операційних систем та розуміння можливостей, які вони дають прикладному програмісту. Виявляти проблеми швидкодії та оптимізувати роботу ПЗ.
Як можна користуватися набутими знаннями і вміннями (компетентності)	<ul style="list-style-type: none"> <li>• сформувати уявлення про внутрішність операційних систем та розуміння можливостей, які вони дають прикладному програмісту.</li> <li>• Створення оптимізованого ПЗ з урахуванням особливостей операційних систем.</li> </ul> Знання архітектури операційних систем будуть корисними при розробці широкого спектру ПЗ.
Інформаційне забезпечення	Силабус дисципліни, презентації лекційного матеріалу, онлайн-документація та альтернативні онлайн лекції
Форма проведення занять	Лекції, лабораторні роботи (комп'ютерні практикуми), самостійна робота.
Семестровий контроль	Залік
Викладач	ст. викладач Ковтунець О.В.

Дисципліна	Технології програмування на мові Python
Кафедра	Обчислювальної техніки
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 6 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку	Базові знання з дисциплін:



<b>вивчення</b>	Програмування, об'єктно-орієнтоване програмування, алгоритми та методи обчислень, системне програмування, алгоритми і структури даних, інженерія програмного забезпечення, основи паралельного програмування
<b>Що буде вивчатися</b>	Методи перетворення проектної документації в реальний код. Способи якими оптимізації процесу написання коду для того щоб виконати поставлені задачі вчасно trade-off стратегії, робота з ризиками проекту і тд. Навички розподілення задач в часі та планування повного циклу розробки. Тестування коду Розгортання коду на серверній інфраструктурі.
<b>Чому це цікаво/треба вивчати</b>	Однією з проблем сучасної розробки ПО є необхідність не тільки якісно писати код але й вміння витримувати строки без сильних втрат якості. Також необхідно вміти працювати як з ризиками, пріоритетами і тд.
<b>Чому можна навчитися (результати навчання)</b>	Отримані знання дозволять навчитися перетворювати проектну документацію в реальний код проекту дотримуючись вимог як по часу так і по якості коду. Роботі в команді розробки.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Розробляти ПО в команді або самостійно.
<b>Інформаційне забезпечення</b>	Силабус дисципліни, навчально-методичний комплекс.
<b>Форма проведення занять</b>	Лекції, практичні та семінарські заняття, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	ст. викладач Шевело О.П.

<b>Дисципліна</b>	<b>Розподілені інформаційні системи</b>
<b>Кафедра</b>	Обчислювальної техніки
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	3 курс, 6 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: Інженерія програмного забезпечення, Програмування, Об'єктно-орієнтоване програмування, Алгоритми та методи обчислень, Системне програмування, Алгоритми і структури даних, Вступ до операційної системи Linux, Архітектура комп'ютерів, Дискретна математика
<b>Що буде вивчатися</b>	— архітектурні шаблони реалізації розподілених інформаційних систем, основи сервіс-орієнтованої архітектури, принципи та протоколи взаємодії веб-сервісів, принципи побудови програмних інтерфейсів, способи інтеграції розподілених програмних ресурсів;

	<ul style="list-style-type: none"> <li>— основи програмної взаємодії з SQL- та NoSQL-базами даних, концепції потокового опрацювання даних;</li> <li>— основи технологічного оснащення процесу проектування, документування, розроблення та тестування програмного забезпечення;</li> <li>— основи побудови веб-застосунків (SinglePageApplication, SPA, PowerWebApplication, PWA) з використанням сучасних програмних засобів (React, Angular, Vue);</li> <li>— способи реалізації процесу збирання та розміщення програмних систем з використанням хмарних сервісів PAAS.</li> </ul>
<b>Чому це цікаво/треба вивчати</b>	— сьогодні володіння знаннями, вміннями і навичками, пов'язаними з процесом проектування розподілених систем колективного використання, процесом розроблення відповідного програмного забезпечення та організації технологічної підтримки всіх стадій життєвого циклу програмного забезпечення, є однією з ключових конкурентних переваг на ринку праці в сфері ІТ.
<b>Чому можна навчитися (результати навчання)</b>	— виконувати проектування інформаційних систем на основі використання сучасних архітектурних принципів із застосуванням засобів моделювання та документування на всіх стадіях життєвого циклу програмного забезпечення, розробляти інформаційне та програмне забезпечення таких систем та організовувати технологічну підтримку процесу їх розроблення та розміщення.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	— виконувати проектування та розроблення сучасних інформаційних систем колективного використання, організовувати технологічну підтримку процесу розроблення програмного забезпечення, використовувати засоби для організації розподіленого оброблення даних, використовувати реляційні, документо-орієнтовані та графові бази даних, інтегрувати власне програмне забезпечення з загальнодоступними веб-сервісами, здійснювати розміщення програмного забезпечення в хмарному середовищі PAAS.
<b>Інформаційне забезпечення</b>	Силабус дисципліни, навчально-методичний комплекс.
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., доцент Болдак А. О.

Дисципліна	<b>DevOps: стратегії розгортання програмного забезпечення.</b>
<b>Кафедра</b>	Обчислювальної техніки
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс</b>	4, семестр 8
<b>Обсяг</b>	4 кредити ЄКТС
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: Програмування, Архітектура комп'ютерів, Комп'ютерна логіка, Інженерія програмного забезпечення,

	Програмування, Алгоритми і структури даних, Бази даних, Linux, Git.
<b>Що буде вивчатися</b>	Курс призначений для вивчення різних підходів до розгортання систем, таких як IaaS, PaaS, SaaS (Kubernetes) і інші. Учасники курсу отримають глибоке розуміння переваг і недоліків кожного підходу та дізнаються, коли і де найкраще використовувати їх у практиці. Програма курсу передбачає практичні роботи реальними хмарними провайдерами. Зміст дисципліни враховує кращі практики від провідних ІТ-підприємств та передбачає участь менторів-практиків для максимального практичного віддзеркалення навчання.
<b>Чому це цікаво/треба вивчати</b>	Широкий обсяг матеріалу: Курс розглядає різні підходи до розгортання систем, охоплюючи IaaS, PaaS, SaaS і інші технології. Студенти отримають повну картину різноманітних методів розгортання програмного забезпечення. Практичні вправи на реальних хмарних провайдерах: Можливість взаємодії з реальними хмарними платформами дозволяє студентам набувати практичний досвід і розвивати навички, які будуть корисні в реальній роботі. Акцент на практичних навичках: Зміст курсу розроблено так, щоб враховувати найкращі практики сучасних підходів розробки і розгортання систем. Це надає студентам можливість набутти практичних навичок, які вони можуть застосовувати в реальних умовах роботи.
<b>Чому можна навчитися (результати навчання)</b>	Цей курс створений для того, щоб студенти могли ефективно використовувати набуті навички в реальних умовах роботи та розвивати себе як кваліфіковані спеціалісти в області розробки та розгортання програмного забезпечення.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Навички які безпосередньо застосовуються при розробці програмного забезпечення
<b>Інформаційне забезпечення</b>	Силабус, РСО, конспект лекцій, завдання на лабораторні роботи
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	Доц. Волокита Артем Миколайович, Solution/Technical Architect DataArt, Зоткін К.В.

## Освітній компонент 7Ф-каталогу

Дисципліна	Обробка та аналіз текстових даних на Python
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 150
Курс, семестр	3 курс, 6 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: програмування, алгоритми та структури даних, математичний аналіз, дискретна математика, теорія ймовірностей. Базові знання мови Python
Що буде вивчатися	популярні бібліотеки мови Python для обробки і аналізу текстових даних, а саме: NLTK, Textblob, GenSim, SpaCy, Scikit-learn; основи роботи з текстами в Python; основні задачі обробки природної мови (NLP) та їх виконання в NLTK та Textblob; визначення тематики текстів в GenSim; машинне навчання та класифікація текстів з SpaCy та Scikit-learn.
Чому це цікаво/треба вивчати	Величезні обсяги даних представлені в текстовому неструктурованому та напівструктурованому форматі, наприклад, лог-файли, статті, відгуки користувачів, пости у соцмережах. То ж існує потреба у видобутку інформації з тексту і використанні її для визначення трендів, настроїв, побудови діалогових систем і т.д. Python є популярною мовою програмування і має низку зручних бібліотек для обробки текстових даних.
Чому можна навчитися (результати навчання)	як обробляти тексти в Python; як використовувати бібліотеки Python для попередньої обробки текстових даних; як візуалізувати текстові дані; як виконувати основні завдання обробки природної мови; як класифікувати тексти; як використовувати глибоке навчання та NLP для створення чат-ботів
Як можна користуватися набутими знаннями і вміннями (компетентності)	створювати програми мовою Python для обробки текстових даних, аналізу системних логів, патентного пошуку, класифікації документів, чат-ботів і т.д.
Інформаційне забезпечення	Силабус
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	ст. викладач Тимофєєва Ю.С.

Дисципліна	Криміналістика в програмно-орієнтованих системах
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 6 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з операційних систем, комп'ютерних мереж, безпеки бездротових і мобільних мереж (не критично, можна одночасне вивчення)
Що буде вивчатися	Основи комп'ютерної криміналістики (Forensics), основні методи та інструменти пошуку інформації по відкритим джерелам (OSINT), основні методи та інструменти криміналістики даних на локальній машині (Computer forensics), у мережі (Network forensics), на мобільних пристроях (Mobile device forensics), криміналістичного аналізу даних (Forensic data analysis),
Чому це цікаво/треба вивчати	<ol style="list-style-type: none"> <li>1. Демонструє типовий підхід до проведення розслідування з комп'ютерної криміналістики.</li> <li>2. Дає можливість самостійно вивчити технології Forensics.</li> <li>3. Надає можливість практичного використання отриманих знань базових дисциплін.</li> <li>4. Користується попитом у роботодавців, підтримується інформаційно з боку компанії-розробника.</li> <li>5. Виводить студента на початковий професійний рівень.</li> </ol>
Чому можна навчитися (результати навчання)	<ol style="list-style-type: none"> <li>1. Основні поняття комп'ютерної криміналістики і правового забезпечення розслідувань інцидентів інформаційної безпеки.</li> <li>2. Основи знань про алгоритми розслідувань інцидентів інформаційної безпеки, проведенні комп'ютерно-технічної експертизи.</li> <li>3. Основні програмні засоби пошуку даних, що використовуються у Forensics.</li> </ol>
Як можна користуватися набутими знаннями і вміннями (компетентності)	<ol style="list-style-type: none"> <li>1. Приймати рішення про те, які методи та інструменти використовувати під час проведення розслідування з комп'ютерної криміналістики.</li> <li>2. Використовувати інструменти криміналістики даних (інструменти аналізу файлів, інструменти аналізу реєстру, інструменти аналізу Інтернет, інструменти аналізу електронної пошти, інструменти аналізу мобільних пристроїв, мережні інструменти комп'ютерної криміналістики тощо).</li> <li>3. Використовувати інструменти OSINT (пошук інформації по відкритим джерелам).</li> <li>4. Самостійно вивчати супутні технології комп'ютерної криміналістики.</li> </ol>
Інформаційне забезпечення	Силабус дисципліни, методичні рекомендації до виконання лабораторних робіт
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	к.т.н., доцент Бойко О.В.

Дисципліна	Бази даних безпілотних систем та автономної робототехніки
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 6 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	<b>Базові знання мови Python</b> , знання HTML+CSS, базові знання теорії баз даних, SQL, знання протоколів World Wide Web.
Що буде вивчатися	Архітектура програмного забезпечення дронів і мобільних роботів. Інформаційна модель навколишнього середовища мобільного робота. Структура інформаційного забезпечення мобільних роботів. Особливості використання в інформаційному забезпеченні мобільних роботів сучасних СУБД (реляційних PostgreSQL, MySQL, SQLite і NoSQL на прикладі MongoDB та ін.) . Використання ORM для спрощення обробки інформації на прикладі ORM Python Peewee та ORM SQLAlchemy. Реалізація доступу до даних СУБД та побудова інтерфейсу користувача інформаційної системи з допомогою фреймворків Flask, Django і концепції MVC. Поняття про ГІС (геоінформаційні системи) та їх використання для цілей навігації, локалізації, орієнтації в просторі та картографування. Збереження та конвертація інформації з використанням XML, JSON, CSV.
Чому це цікаво/треба вивчати	Python — сучасна універсальна мова програмування і є найпоширенішою в вбудованих системах управління в мобільній робототехніці і її можна використовувати для реалізації будь-якого класу задач від простого сценарію автоматизації до системного програмування, від рішення задач навігації та локалізації з використанням ГІС (геоінформаційної системи робота) до програмування задач управління мікроконтролерами та мікрокомп'ютерами дрона або робота, використання фреймворків Python для зв'язку з структурованою інформацією, яка потребує для своєї обробки різних типів БД (реляційних, NoSQL, графових, табличних і інших). Для обробки інформації і обміну з зовнішнім світом роботів потребує розвинутої серверної частини, побудованої з використанням протоколів і стандартів WWW. Розробка серверної частини без фреймворку втомлює, тому фреймворки і закріпилися в якості основи веб-розробки мобільних роботів. Одним з найбільш поширених в світі є Python фреймворки Django, Flask.
Чому можна навчитися (результати навчання)	Розробляти інформаційне забезпечення мобільних роботів з користанням мови Python та СУБД різних типів та архітектури, користуватися для прискорення розробки ORM; використовувати засоби автоматизації та фреймворки для розробки інформаційного забезпечення для безпілотних та автономних систем для серверної частини системи управління роботом;
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> <li>• для розробки систем управління автономними мобільними роботами та дронами;</li> <li>• для вирішення задач навігації та локалізації для інших безпілотних систем.</li> </ul>

Інформаційне забезпечення	Силабус дисципліни, навчально-методичний комплекс.
Форма проведення занять	Лекції, практичні та семінарські заняття
Семестровий контроль	Залік
Викладач	к.т.н., доцент Резніков С.А.

Дисципліна	Сучасні технології розробки WEB-застосунків з використанням мови Python
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 6 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: основ програмування, реляційні бази даних та основи SQL (можливе одночасне вивчення), основи HTML та CSS (можливе одночасне вивчення). Бажаю основи JavaScript (не критично).
Що буде вивчатися	Способи аналізу функціональних вимог для сучасних Web-додатків; - створення проекту і Web-застосунків на основі Django; - розробка моделі об'єктно-реляційної проекції баз даних Django; - розробка системи Web-інтерфейсів користувача на основі Django, у тому числі з використанням Bootstrap та інших JavaScript фреймворків; - система тестування для Web-застосунків на основі Django; - мікроархітектура на основі Django Web-застосунків; - розгортання і контейнеризація Django Web-застосунків на Docker.
Чому це цікаво/треба вивчати	За один семестр можна вивчити і отримати перший професійний досвід розробки Web-застосунків на основі Django. Цікаво для тих, хто маючи знання основ програмування, HTML та CSS бажає вийти на професійний рівень розробки Web-застосунків і планує після університету працювати в цій сфері, або почати свій бізнес. Тема курсу розрахована на опанування наступних елементів розробки Web-застосунків: розробка Web-інтерфейсів користувача, серверної частини Web-застосунків, об'єктно-реляційної проекції баз даних, тестування Web-застосунків, бізнес-аналітики, документування, менеджменту, тощо.
Чому можна навчитися (результати навчання)	Проектуванню і бізнес-аналітики Web-застосунку на основі Django. Повному циклу розробки Web-застосунку на основі Django. Розробці мікроархітектури на основі Django Web-застосунків і розгортанню програмної системи на основі контейнеризації. Професійному тестуванню Web-застосунку на основі Django.
Як можна користуватися набутими знаннями і вміннями	Отриманні знання, вміння, досвід і навички дають можливість працевлаштування у відповідній професійній сфері. Також отриманий досвід дозволяє зробити перший крок щодо

(компетентності)	створення власного бізнесу на основі Web-застосунків і програмних систем.
Інформаційне забезпечення	Силабус, методичні рекомендації до виконання лабораторних робіт, інформаційна підтримка у Телеграмі (група відкривається на час занять), середовище розробки ( <a href="https://www.jetbrains.com/ru-ru/pycharm/">https://www.jetbrains.com/ru-ru/pycharm/</a> ), довідкова система ( <a href="https://www.djangoproject.com/start/">https://www.djangoproject.com/start/</a> ).
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	к.т.н., доцент Резніков С.А.

Дисципліна	Комп'ютерна графіка та обробка зображень
Кафедра	Інформатики та програмної інженерії
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 6 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання лінійної алгебри та аналітичної геометрії (вектори, матриці, векторні рівняння). Базові знання з інтегрального та диференційного числення. Основи програмування, базові знання ООП.
Що буде вивчатися	Створення зображень з 3d моделей методом трасування променів (path tracing): рівняння рендерингу, перетин променів із тривимірними примітивами, моделі затінення, матеріали, BRDF, метод Монте-Карло. Realtime rendering: базова архітектура графічних процесорів, shaderpipeline, представлення тривимірних моделей, растеризація, текстуровання, матеріали, geometry/vertex/pixel shaders, PBR, пост-обробка. Обробка зображень: типові алгоритми перетворень, корекція кольору, tonemapping, формати/методи стискання зображень.
Чому це цікаво/треба вивчати	Цей курс має на меті познайомити слухача з тим, як працюють речі, із якими більшість із нас щоденно стикається. Прикладом таких речей можуть бути тривимірні графіки в сучасних іграх, комп'ютерні спецефекти у фільмах або просто зображення на моніторі. Навіть текст, який ви зараз читаете, надрукований шрифтом, що складається з кривих Без'є і є прикладом векторної графіки. Комп'ютерна графіка це область ІТ, в якій не просто оброблюються байти, передаються з сервера на сервер невидимі пакети, і не вщухають суперечки між прихильниками ООП та ФП. Натомість ви і будь-хто, навіть не розбираючись в темі, може побачити на власні очі результати роботи вашого коду.
Чому можна навчитися (результати навчання)	Крім насамперед створення реалістичних зображень та обробки тривимірної графіки слухачі курсу матимуть змогу навчитися використовувати багатопоточність для розпаралелювання складних обчислювальних задач, писати доволі складні програми з монолітною архітектурою; познайомитися на практиці з ООП та



	SOLIDтарозширити коло знайомих алгоритмів (MC integration, spatial access methods, Huffman coding, LZW)
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Використовувати отримані знання для розробки комп'ютерних ігор або рушіїв комп'ютерних ігор.
<b>Інформаційне забезпечення</b>	Навчальна та робоча програми дисципліни, PCO, контрольні завдання
<b>Форма проведення занять</b>	Лекції та лабораторні (комп'ютерні практикуми)
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., с.н.с. Щебланін Ю.М.

<b>Дисципліна</b>	<b>Комп'ютерна графіка та мультимедіа</b>
<b>Кафедра</b>	Інформатики та програмної інженерії
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	3 курс, 6 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	навички програмування на мові високого рівня; знання лінійної алгебри, геометрії та тригонометрії; розв'язування рівнянь, похідних, інтегралів; знання векторів, матриць; вміння розв'язувати системи рівнянь.
<b>Що буде вивчатися</b>	<ul style="list-style-type: none"> <li>• вступ до комп'ютерної графіки;</li> <li>• програмування 2D графіки;</li> <li>• трансформація та відображення об'єктів 3D графіки;</li> <li>• моделювання 3D об'єктів та знаходження видимої поверхні;</li> <li>• освітлення та рендеринг поверхні;</li> </ul> алгоритми трасування променів та їх застосування.
<b>Чому це цікаво/треба вивчати</b>	отримані знання та набуті практичні навички дозволяють розуміти принципи створення та редагування цифрових зображень, а також комп'ютерних ігор.
<b>Чому можна навчитися (результати навчання)</b>	<ul style="list-style-type: none"> <li>• пояснювати основні концепції комп'ютерної графіки, включаючи перегляд, проекцію, перспективу, моделювання та перетворення у двох та трьох вимірах;</li> <li>• застосовувати концепції колірних моделей, моделей освітлення та затінення, текстур, трасування променів, прихованої поверхні, згладжування та рендерингу;</li> <li>• інтерпретувати математичні засади комп'ютерної графіки;</li> <li>• описувати основи анімації, параметричні криві та поверхні, а також точкове освітлення;</li> <li>• визначати типовий графічний конвеєр та застосувати методи графічного програмування для створення комп'ютерної графіки; створювати ефективні програми для вирішення задач програмування</li> </ul>

	комп'ютерної графіки, включаючи 3D-трансформацію, моделювання об'єктів, колірне моделювання, освітлення, текстури та трасування променів.
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>створювати цифрові зображення та інтерактивну 3D графіку;</li> <li>розробляти комп'ютерні ігри та додатки.</li> </ul>
<b>Інформаційне забезпечення</b>	Навчальна та робоча навчальна програма дисципліни, РСО, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
<b>Форма проведення занять</b>	Лекції та лабораторні (комп'ютерні практикуми)
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., доцент Родіонов П.Ю.

<b>Дисципліна</b>	<b>Управління IT-інфраструктурними проектами (Сертифікатна програма, авторський курс компанії Global Logic)</b>
<b>Кафедра</b>	Обчислювальної техніки
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	3 курс, 6 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: Програмування, Архітектура комп'ютерів, Дискретна математика, Комп'ютерна логіка, Інженерія програмного забезпечення, Програмування, Об'єктно-орієнтоване програмування, Алгоритми і структури даних, Бази даних, Linux, Git
<b>Що буде вивчатися</b>	<p>Головна мета дисципліни – набуття теоретичних знань і практичних навичок управління проектами та командної роботи: набуття Soft skills шляхом застосування своїх Hard skills для розв'язку задач практики.</p> <p>Реалізується проєктний підхід як форма організації і реалізації конкретного проєкту в сфері інформаційних технологій та комп'ютерної інженерії.</p> <p>Зміст дисципліни враховує найкращі практики провідних IT-підприємств і передбачає залучення менторів – практиків.</p> <p>Результатом опрацювання завдань дисципліни є завершений практичний проєкт.</p> <p>Дисципліна передбачає вивчення взаємопов'язаних компонент:</p> <p>I. Менеджмент проєктів:</p> <p>організація та реалізація: допроєктного R&amp;D етапу; розробка технічної пропозиції і технічного завдання; інженерія вимог (бізнес-аналіз); архітектурне проектування; розробка системи (програмного забезпечення); оцінювання якості (верифікація і тестування);</p>

	<p>впровадження і основи DevOps; супроводження; організація процесів (моделей життєвого циклу) за класичними та Agile моделями; звітування і здача проектну.</p> <p>II. Групова динаміка.</p> <ul style="list-style-type: none"> <li>• формування групи і команди;</li> <li>• організація командної роботи;</li> <li>• організація і проведення ефективних нарад, презентацій, переговорів.</li> </ul> <p>III. Практична частина:</p> <ul style="list-style-type: none"> <li>• Git/Gitea;</li> <li>• CI/CD інфраструктура;</li> <li>• Jenkins;</li> <li>• JIRA;</li> <li>• SonarCube;</li> <li>• Творча робота в командах</li> <li>• Презентація досягнень</li> </ul>
<p><b>Чому це цікаво/треба вивчати</b></p>	<p>Курс розроблено за участі фахівців, які працюють на реальних проектах в ІТ-компаніях. Курс розроблено по аналогії з PoC проектом «Smart City», в якому приймали участь студенти факультету <a href="https://comsys.kpi.ua/na-kafedri-obchislyvalnoyi-texniki-fiot-zaprovadjena">https://comsys.kpi.ua/na-kafedri-obchislyvalnoyi-texniki-fiot-zaprovadjena</a>. Все як в реальному проекті – Git, Jira, SonarCube, базові навички Devops, підняття CI/CD інфраструктури, перший коміт та перший мердж, творча робота в командах над проектом та презентація досягнень. Купа Soft Skills, навички використання сучасних проектних технологій на практиці і сподівання, що незабаром і на нашій карті «Smart City» поїдуть машинки та замиготять світлофори.</p> <p>Курс «Управління ІТ-інфраструктурними проектами» може бути прослуханий студентами як окремий курс, так і у складі <b>сертифікатної програми «Інженерія вбудованих систем та Інтернет речей»</b>, яка запроваджена на факультеті інформатики та обчислювальної техніки з 2021 – 2022 навчального року. Детально з сертифікатною програмою можна ознайомитися на сайті кафедри обчислювальної техніки <a href="https://comsys.kpi.ua/sertifikatni-programi">https://comsys.kpi.ua/sertifikatni-programi</a>.</p> <p>Перелік дисциплін сертифікатної програми:</p> <ul style="list-style-type: none"> <li>• Технології програмування C/Embedded – 5 семестр;</li> <li>• Управління ІТ-інфраструктурними проектами – 6 семестр;</li> <li>• Технології програмування на ПЛІС (FPGA) – 6 семестр;</li> <li>• Тестування та контроль якості (QA) вбудованих систем – 7 семестр;</li> <li>• Технології розроблення вбудованих систем IoT – 8 семестр.</li> </ul>
<p><b>Чому можна навчитися (результати навчання)</b></p>	<p>Знання та навички організації, застосування і реалізації:</p> <ul style="list-style-type: none"> <li>– командного виконання проектів з використанням усіх фаз життєвого циклу (від збору вимог до впровадження);</li> <li>– виконання проектів з використанням різних методологій розробки (класичних та Agile) та на різних ролях (розробник, тестувальник, аналітик, DevOps, керівник тощо);</li> <li>– використання автоматизованих засобів командної розробки проектів;</li> <li>– бізнес-комунікації (наради, презентації, активне слухання, письмова комунікація, проходження співбесіди тощо);</li> <li>– роботи в команді (надавання зворотного зв'язку, ділення досвідом,</li> </ul>

	<p>вирішування конфліктних ситуацій тощо);</p> <ul style="list-style-type: none"> <li>- управління часом;</li> <li>- практичного вирішення проблемних ситуацій;</li> </ul> <p>пошуку, збору та обробки інформації, необхідної для реалізації практичних завдань.</p>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Командна робота над складними технічними проектами відповідно до сучасних вимог до процесів і технологій їх реалізації.
<b>Інформаційне забезпечення</b>	Силабус дисципліни, навчально-методичний комплекс.
<b>Форма проведення занять</b>	Лекції, лабораторні роботи.
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	QA Engeneer GlobalLogic Таранюк В.А., асистент Мінченко Л.Д.

Дисципліна	Паралельні та розподілені обчислення
<b>Кафедра</b>	Обчислювальної техніки
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	3 курс, 6 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: Програмування, Об'єктно-орієнтоване програмування, Алгоритми та методи обчислень, Системне програмування, Алгоритми і структури даних, Архітектура комп'ютерів, Агальтична геометрія, Інженерія програмного забезпечення
<b>Що буде вивчатися</b>	<p>Методи організації обчислень в паралельних комп'ютерних системах з різною структурною організацією: систем зі спільною пам'яттю, систем з локальної пам'яттю, розподілених систем. Життєвий цикл розробки програми для паралельної системи, що включає:</p> <ul style="list-style-type: none"> <li>- розробку паралельного алгоритму обчислень .</li> <li>- розробку алгоритмів виконання потоків з рішенням завдання взаємного виключення та синхронізації.</li> <li>- розробку структурної схеми взаємодії потоків для обраних мов (бібліотек) паралельного програмування та засобів організації потоків: семафорів, мютексів, подій, критичних секцій,</li> <li>- методи налагодження і тестування паралельної програми</li> <li>- дослідження відсотку завантаження процесорів (ядер) системи, визначення часу виконання, обчислення коефіцієнту прискорення (speedup).</li> </ul> <p>В лабораторних роботах використовуються сучасні мови та бібліотеки паралельного програмування: Java, C#, WinAPI, Ada, OpenMP, MPI, TBB.</p>
<b>Чому це цікаво/треба</b>	Сучасні комп'ютери (починаючи від планшету і закінчуючи супер

<b>вивчати</b>	комп'ютерами) обладнано багатоядерними процесорами, тому створення ефективного програмного забезпечення для них потребує використання потоків для оптимального використання багатоядерного обладнання. Багатоядерні процесори також є частиною більшості гаджетів, таких як смартфони, mp3 плеєри, відеоплеєри, ігрові приставки, що дозволяє прискорити (покращити) обробку звуку та зображення.
<b>Чому можна навчитися (результати навчання)</b>	Вмінню розробляти програмне забезпечення для паралельних та розподілених комп'ютерних систем різного призначення, аналізувати завдання на наявність паралелізму, будувати оптимальні паралельні алгоритми, використовувати потоки для побудови програми, організувати оптимальну взаємодію потоків в залежності від структури комп'ютерної системи, виконувати налагодження паралельної програми, запобігати та позбавлятися тупикових ситуацій. Використанню сучасних мов та бібліотек паралельного програмування.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Отримані базові знання допоможуть у подальшому професійному навчанні в тому числі самостійно, розширюють перспективу кар'єрних можливостей в галузі розроблення програмного забезпечення для будь яких паралельних комп'ютерних систем, систем реального часу, а також створення серверно-клієнтських додатків.
<b>Інформаційне забезпечення</b>	Силабус дисципліни, навчально-методичний комплекс.
<b>Форма проведення занять</b>	Лекції, практичні та семінарські заняття, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., доцент Корочкін О.В.

## Освітній компонент 8Ф-каталогу

**Для всіх освітньо-професійних програм ФІОТ, окрім студентів, що навчаються за ОП «Комп'ютерні системи та мережі», спеціальності 123 «Комп'ютерна інженерія»**

Дисципліна	Технології та протоколи мультисервісних мереж
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 6 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: фізика, системотехніка, комп'ютерні мережі, інтернет-технології
Що буде вивчатися	Використання та створення інформаційно-комунікаційних технологій, розповсюдження та використання мультисервісних мереж. Особливості і порядок застосування базових технологій розробки мережевих протоколів на базі стандартів еталонних моделей OSI/ISO та стеків протоколів TCP/IP, IPX/SPX, SS7 (рівні MTP, SCCP, MAP, ISUP), кореляція SS7 з моделлю OSI; знання, розуміння та застосування VoIP/SIP: загальні принципи, callflow; кореляція стеку протоколів TCP/IP з моделлю OSI (призначення протоколів, адресація, маршрутизація, загальні принципи функціонування IP-мереж). Застосування протоколів маршрутизації OSPF, RIP, BGP, EIGRP в мультисервісних мережах. Технології мультисервісних транспортних мереж PDH, SDH, xWDM, MPLS. Особливості застосування технологій доступу xDSL, CATV, PON, FTTx, WiMax. Протоколи IP, HTTP та DHCP для WEB – технологій. DHCPv4 та DHCPv6. Особливості застосування SMPP. Поняття мобільності в стільникових мережах зв'язку. Структури GSM мережі (включно з призначенням та функціями її елементів - HLR, MSC, VLR, SMSC, GMSC, SGSN, GGSN тощо). Структура LTE мережі (архітектура та призначення основних елементів).
Чому це цікаво/треба вивчати	Стрімкий розвиток сучасних інформаційних технологій надає нові можливості для використання на різних рівнях комп'ютерних мереж, мережевих протоколів різної функціональності для розвитку відповідних інформаційних сервісів і організації взаємодії інформаційного простору з їх партнерами та клієнтами, використання мобільного середовища та різномірних джерел інформації і пристроїв комунікації.
Чому можна навчитися (результати навчання)	Знання та розуміння моделі OSI/ISO. Знання, розуміння та застосування стеку протоколів SS7. Знання, розуміння та застосування VoIP/SIP. Знання, розуміння та застосування стеку протоколів TCP/IP. Знання, розуміння та застосування PDH, SDH, xWDM, MPLS. Знання, розуміння та застосування SMPP та SIGTRAN (основні рівні, що відповідають за передачу трафіку та їх основні мітки). Знання та розуміння поняття мобільності в стільникових мережах зв'язку та яким чином вона забезпечується. Знання та розуміння структури GSM та LTE мереж.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Здатність до проектування, розробки, налагодження та вдосконалення системного, комунікаційного та програмно-апаратного забезпечення інформаційних систем та технологій, Інтернету речей (IoT), комп'ютерно-інтегрованих систем та

	<p>системної мережної структури, управління ними.</p> <p>Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).</p> <p>Здатність управляти якістю продуктів і сервісів інформаційних систем та технологій протягом їх життєвого циклу.</p> <p>Здатність вибору, проектування, розгортання, інтегрування, управління, адміністрування та супроводжування інформаційних систем, технологій та інфокомунікацій, сервісів та інфраструктури організації.</p> <p>Здатність управляти та користуватися сучасними інформаційно-комунікаційними системами та технологіями(у тому числі такими, що базуються на використанні Інтернет).</p>
<b>Інформаційне забезпечення</b>	Силабус дисципліни, електронний конспект та презентація
<b>Форма проведення занять</b>	Лекції, комп'ютерний практикум
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	д.т.н., професор Жураковський Б.Ю.

<b>Дисципліна</b>	<b>Програмні засоби проектування та реалізації нейромережових систем</b>
<b>Кафедра</b>	Інформаційних систем та технологій
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 150
<b>Курс, семестр</b>	3 курс, 6 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: Інтелектуальні системи та технології, програмування мовою Python
<b>Що буде вивчатися</b>	Проектування, реалізація та дослідження програмних систем з використанням штучного інтелекту. Методи створення, навчання і функціонування нейромережових систем та систем глибокого навчання. Програмні засоби реалізації та дослідження нейромережових систем на Python з використанням бібліотек TensorFlow і Keras.
<b>Чому це цікаво/треба вивчати</b>	Машинне навчання – розділ інформатики, який вивчає розробку алгоритмів, які можна навчати. Одним з розділів машинного навчання є штучні нейронні мережі. Глибоке навчання є однією з найгарячіших галузей науки про дані, оскільки багато тематичних досліджень дають вражаючі результати в робототехніці, розпізнаванні зображень та штучному інтелекті. Однією з найпотужніших і простих у використанні бібліотек Python для розробки та оцінки програмних моделей глибокого навчання є Keras. Він обгортає ефективні чисельні бібліотеки обчислень Theano

	та TensorFlow. Вивчення даних програмних технологій реалізації та дослідження нейромережових систем дозволить легко та простореалізовувати складні програмні системи штучного інтелекту.
<b>Чому можна навчитися (результати навчання)</b>	При вивченні даної дисципліни, студенти вивчать теоретичні основи нейронних мереж та отримають початковий досвід в розробці програмного забезпечення, що реалізує нейромережові технології. На лабораторних заняттях отримають початковий досвід створення програмних систем на основі нейронних мереж. Проведуть розробку та дослідження нейромережових систем різного призначення за допомогою мови програмування Python та бібліотек TensorFlow і Keras.
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	Набуті знання можна використати при розробці програмних систем зі штучним інтелектом.
<b>Інформаційне забезпечення</b>	Силабус, конспект лекцій, презентаційні матеріали, методичні вказівки до виконання лабораторних робіт
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., доцент Шимкович В.М.

<b>Дисципліна</b>	<b>Професійне використання SQL та PL/SQL на прикладі РСУБД Oracle 11g</b>
<b>Кафедра</b>	Інформаційних систем та технологій
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 150
<b>Курс, семестр</b>	3 курс, 6 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з теорії реляційних баз даних та проектування реляційних баз даних в розділі інфологічного та фізичного моделювання. Основи роботи в SQL, мінімальне адміністрування та налаштування серверу OracleDatabase 11gXE.
<b>Що буде вивчатися</b>	В курсі буде вивчатися PL/SQL в розділах створення збережуваних процедур, функцій, тригерів, динамічний SQL та PL/SQL. Більш глибоко відпрацюємо запити на виборку, складні запити, аналіз та оптимізація запитів.
<b>Чому це цікаво/треба вивчати</b>	При створенні програмних застосунків важливе місце відводиться саме базі даних, написанню скриптів та запитів для коректної, швидкої роботи застосунків. PL/SQL одна з найуживаніших мов на рівні T-SQL, PL/pgSQL та інші. Знання базових конструкцій підвищить конкурентоспроможність розробника серед інших претендентів.
<b>Чому можна навчитися (результати навчання)</b>	Ознайомитись і навчитись писати складні конструкції мови PL/SQL, аналізувати запити DML, застосовувати PL/SQL для певного



	програмного застосування.
Як можна користуватися набутими знаннями і вміннями (компетентності)	Набуті знання можна застосовувати при проектуванні WEB застосувань, адмініструванні реляційних серверів баз даних, програмування мовою PL/SQL.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, РСО, методичні вказівки з виконання лабораторних робіт (планується), конспект лекцій (планується), презентаційний матеріал та файли з прикладом скриптів (планується).
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	к.т.н., доцент Ульяницька К.О.

Дисципліна	Сучасні технології розробки WEB-застосувань на платформі .NET (2 частина)
Кафедра	Інформатики та програмної інженерії
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 30 до 60
Курс, семестр	3 курс, 6 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	<p><u>Обов'язково:</u></p> <ol style="list-style-type: none"> <li>1. Знання синтаксису мови C#, і основних принципів роботи платформи .NET;</li> <li>2. Розуміння SOLID та OOP;</li> <li>3. Патерни проектування</li> <li>4. Розуміння принципів роботи веб-застосунку та протоколів TLS, SSL, HTTP(s);</li> <li>5. Розуміння LayerArchitecture;</li> <li>6. Знання ORM (EntityFramework).</li> </ol>
Що буде вивчатися	<ol style="list-style-type: none"> <li>1. Архітектурні підходи до створення веб-застосунків та їх порівняння</li> <li>2. Способи оптимізації веб-застосунків</li> <li>3. Автоматизоване тестування веб-застосунків в межах піраміди тестування</li> </ol>
Чому це цікаво/треба вивчати	Аналіз пропозицій працевлаштування в ІТ-галузі на загальновідомих інтернет-агрегаторах дає змогу зрозуміти, що розробники програмного забезпечення з використанням платформи .NET займають одні з перших позицій щодо попиту на їх знання та вміння. Компетенції, що розглядаються в межах даного курсу є складовими частинами більшості вимог до кандидатів в межах вакансій на back-end або full-stack розробника.
Чому можна навчитися (результати навчання)	<ol style="list-style-type: none"> <li>1. Використовувати різні архітектури для вирішення практичних завдань;</li> <li>2. Виконувати задачі оптимізації веб-застосувань на різних архітектурних рівнях.</li> <li>3. Використовувати автоматизовані підходи до тестування веб-</li> </ol>

	застосувань. 4. Виконувати рефакторинг програмного коду.
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	Дисципліна надає набір вмінь та компетенцій, що є мінімально необхідними для спеціаліста з розробки програмного забезпечення, який бажає стати C#/ .Net-розробником. Компетенції, які надає дисципліна: <ul style="list-style-type: none"> <li>● Створення веб-застосунків на мові C#, що працюють на платформах сімейства .NET;</li> <li>● Вміння застосовувати різні архітектурні підходи для оптимального вирішення поставлених задач.</li> <li>● Вміння виконувати оптимізації веб-застосунків</li> <li>● Вміння виконувати рефакторинг та тестування веб-застосунків</li> </ul>
<b>Інформаційне забезпечення</b>	Силабус дисципліни, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
<b>Форма проведення занять</b>	Лекції та лабораторні (комп'ютерні практикуми)
<b>Семестровий контроль</b>	Залік, 6-й семестр
<b>Викладач</b>	ас. Вовк Є.А.

<b>Дисципліна</b>	<b>Методи та системи штучного інтелекту</b>
<b>Кафедра</b>	Інформатики та програмної інженерії
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	3 курс, 6 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Повний курс дискретної математики, повний курс вищої математики та теорії ймовірності
<b>Що буде вивчатися</b>	Що таке штучний інтелект : Історія ШІ , Тест Тьюрінга та чаттерботів, Розподілені, автономні та навчальні агенти. Служба робототехніки Системи на основі знань: Логіка предикатів першого порядку, Обмеження логіки, Проблема простору пошуку, Моделювання невизначеності Пошук, ігри та вирішення проблем: Неінформований пошук, Евристичний пошук Ігри з супротивниками. Функції евристичного оцінювання Обґрунтування невизначеності: Обчислення з ймовірностями, Принцип максимальної ентропії, Розум з байєсівськими мережами
<b>Чому це цікаво/треба вивчати</b>	Штучний інтелект універсальний ШІ не обмежується лише компютерною та космічною галузями, але також відіграє значну роль у галузях, які безпосередньо пов'язані зі звичайною людиною, як сфера охорони здоров'я, автомобілебудування та навіть банківський та фінансовий сектор. Вона застосовна для будь-якої галузі, і ніколи не дає результатів. Майстерність століття Безперечно коли-небудь AI замінить багато робочих місць, а також створить багато і різних можливостей для роботи в суміжних галузях.

	Для того, щоб бути сучасним, кожен повинен мати хоча б якусь інформацію про AI. Покращене управління катастрофами Коли стихійне лихо вражає регіон чи державу, часто люди, які проживають там, звертаються до соціальних медіа з проханням про допомогу, окрім допомоги, яку отримують від урядових організацій. Багато разів жертви таких катастроф записують відео та діляться ними на платформах соціальних медіа, таких як Facebook та Twitter. На цих платформах є програми з підтримкою ШІ, які служать носієм для поширення новин про ці катастрофи.
<b>Чому можна навчитися (результати навчання)</b>	Створювати системи штучного інтелекту для вирішення прикладних задач із пошуком та логічним виводом
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>• Створювати системи штучного інтелекту з нуля</li> <li>• Вибирати коректні підходи до створення систем штучного інтелекту</li> </ul> Проводити аналіз якості роботи систем штучного інтелекту та визначати коректність їх роботи
<b>Інформаційне забезпечення</b>	Силабус дисципліни, контрольні завдання, підручник
<b>Форма проведення занять</b>	Лекції та лабораторні (комп'ютерні практикуми)
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., доцент Курченко О.А.

Дисципліна	Цифрова обробка сигналів
<b>Кафедра</b>	Обчислювальної техніки
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	3 курс, 6 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська, англійська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: Вища математика, Програмування, Дискретна математика, Архітектура комп'ютерів, Комп'ютерні системи
<b>Що буде вивчатися</b>	Математичні основи та алгоритми цифрової обробки сигналів (ЦОС) і зображень, а також методи їх складання, програмування та апаратної реалізації.
<b>Чому це цікаво/треба вивчати</b>	Сучасні апаратні та програмні засоби, що поширені в ІТ-галузі дуже часто ґрунтуються на виконанні алгоритмів ЦОС. Це, наприклад, засоби автоматизації, телекомунікації, обробки зображень, комп'ютерні ігри, розпізнавання образів та штучний інтелект, IoT. Без знань алгоритмів ЦОС практично неможливо тестувати, модернізувати та розробляти такі засоби.
<b>Чому можна навчитися (результати навчання)</b>	— вибирати, моделювати та модернізувати алгоритми ЦОС; — розробляти застосунки, які виконують алгоритми ЦОС.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	— проводити аналіз, діагностування, вибір існуючих засобів ЦОС, — виконувати модернізацію існуючих та розробку нових засобів ЦОС,

	— впроваджувати алгоритми ЦОС у засобах IoT та штучного інтелекту. Знати основи ЦОС та мати досвід її програмування — це обов'язкова вимога при прийомі на роботу у більшості фірм, що займаються розробкою та поширенням застосунків та апаратури для автоматизації, телекомунікації, штучного інтелекту, IoT.
Інформаційне забезпечення	Силабус дисципліни, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	д.т.н., професор Сергієнко А.М.

Дисципліна	Проектування та реалізація операційних систем
Кафедра	Обчислювальної техніки
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	<b>Окрім студентів каф. ОТ</b> Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 6 семестр
Обсяг	120 годин / 4 кредити ЄКТС
Мова викладання	Українська
Вимоги до початку вивчення	Розуміння Асемблера, розуміння С або С++ для читання прикладів у лекціях. Базові знання з дисциплін Програмування, Структури даних та алгоритми, Архітектура комп'ютерів, Системне програмування.
Що буде вивчатися	Вивчатимуться принципи функціонування, архітектура та реалізація ядер операційних систем загального призначення. За основу беруться реалізації ядер Unix-подібних операційних систем та реалізації інших типів ядер, тобто дисципліна не зосереджена на якійсь одній конкретній операційній системі. Надається ґрунтовна інформація про задачі та способи їх вирішення в ядрі операційної системи на низькому рівні реалізації. Дисципліна складається з наступних розділів: ядро та процеси, підтримка багатопотокових програм, файлові системи, управління пам'яттю.
Чому це цікаво/треба вивчати	Цю дисципліну доцільно вивчати тим, хто буде розробляти ядра або частини ядер операційних систем. Ця дисципліна також корисна системним програмістам та прикладним програмістам для поглибленого розуміння функціонування операційної системи, що дозволить розробляти ефективніші програми.
Чому можна навчитися (результати навчання)	Підготуватися до розробки та розуміння вихідного коду частин ядер операційних систем загального призначення що реалізують управління процесами та потоками, системні виклики, файлові системи, управління пам'яттю.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Набуті знання можливо використовувати при розробці ядер операційних систем, для супроводження вихідного коду наявних ядер операційних систем, при розробці ефективних системних та прикладних програм.
Інформаційне забезпечення	Силабус, конспект лекцій, завдання на лабораторні роботи

Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	Ст. викладач Сімоненко Андрій Валерійович

Дисципліна	Системне програмування на Асемблері
Кафедра	Обчислювальної техніки
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс	3-й, 6 семестр
Обсяг	4 кредити ECTS
Мова викладання	Українська, англійська
Вимоги до початку вивчення	Базові знання з архітектури комп'ютерів та основ програмування. Бажано мати уявлення про операційні системи. Корисним буде також знання мови програмування C, розуміння структур даних та алгоритмів. Бажано (необов'язково) мати досвід роботи з якимось середовищем розробки програмного забезпечення.
Що буде вивчатися	<p>Базові поняття:</p> <ul style="list-style-type: none"> <li>- Базове середовище виконання програм на рівні процесора, зокрема, організація пам'яті, регістри процесора, операнди команд та способи адресації операндів, стек. Особливості побудови та використання 32-х та 64-бітових процесорів.</li> <li>- Побудова та компіляція програм на Асемблері. Основні шаблони програмного коду. Інструментальні засоби розробки та налагодження програм.</li> <li>- Типи та формати даних. Стандарт IEEE754 та інші</li> <li>- Система команд процесорів сімейства x86 та сумісних з ними</li> </ul> <p>Просунуті аспекти:</p> <ul style="list-style-type: none"> <li>- Програмування даних підвищеної (довільної) розрядності та операцій з ними</li> <li>- Програмування викликів системних функцій</li> <li>- Створення власних бібліотек</li> <li>- Створення проектів на C/C++ з використанням модулів на Асемблері. Конвенції виклику процедур на Асемблері</li> <li>- Використання розширень системи команд для підвищення швидкодії програмного забезпечення. Програмування з використанням AVX, AVX2, 512, FMA та інших розширень.</li> <li>- Особливості програмування 64-бітових процесорів.</li> </ul> <p>Середовища розробки програмного забезпечення</p> <ul style="list-style-type: none"> <li>- Microsoft Visual Studio (C/C++, Assembler – проекти 32- та 64-бітових програм)</li> <li>- FASM, NASM та інші</li> </ul>
Чому це цікаво/треба вивчати	<p>Знання та вміння використання програмування на Асемблері дозволяють створювати ефективні програмні засоби, зокрема, в проектах на основі мов програмування високого рівня. Це може забезпечувати найвищу швидкість обробки даних завдяки найповнішому використанню можливостей сучасних процесорів, що може суттєво підвищити конкурентноздатність програмних рішень.</p> <p>Тільки написавши та налагодивши програму на Асемблері</p>

	можна нарешті пізнати архітектуру комп'ютера та зрозуміти як він насправді працює. А це є передумовою набуття високої кваліфікації фахівця-комп'ютерщика.
<b>Чому можна навчитися (результати навчання)</b>	<p>Вивчити архітектуру сучасних процесорів, організацію пам'яті та формати представлення даних. Основні та розширені набори системи команд.</p> <p>Навчитися створювати компактні та ефективні програмні рішення у вигляді проєктів на основі мов C/C++ з використанням модулів на Асемблерів задля досягнення найвищої швидкодії обробки інформації. Створювати власні бібліотеки.</p>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<p>Набуті знання, уміння та практичні навички дозволять успішно реалізовувати складні проєкти у будь-якій галузі – усюди, де необхідно досягти найвищої швидкодії обробки інформації.</p> <p>Майбутній фахівець може використати набуті знання для побудови власних модулів та бібліотек прикладного та системного програмного забезпечення.</p> <p>Набуті компетентності з програмування на рівні процесорів означатимуть глибоке розуміння того, як насправді побудовані та функціонують апаратні та програмні комп'ютерні засоби, що відкриває можливості подальшого підвищення кваліфікації фахівців з ІТ-технологій та зростання конкурентоздатності програмних рішень.</p>
<b>Інформаційне забезпечення</b>	Силабус, навчально-методичний комплекс
<b>Форма проведення занять</b>	Лекції, лабораторні роботи, самостійна робота
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	Доцент, к.т.н. Порєв В.М.

## 4 курс 7 семестр

### Освітній компонент 8.1 Ф-каталогу(для груп ІО)

#### Для груп ІО

123 «Комп'ютерна інженерія»  
ОП «Комп'ютерні системи та мережі»

Дисципліна	Технології та протоколи мультисервісних мереж
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	4 курс, 7 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: математика, фізика, комп'ютерні мережі, інтернет-технології
Що буде вивчатися	Використання та створення інформаційно-комунікаційних технологій, використання алгоритмічних мов та середовищ розробки у процесах проектування, розробки, розповсюдження та використання мультисервісних мереж. Особливості і порядок застосування базових технологій розробки мережевих протоколів на базі стандартів еталонних моделей OSI/ISO та стеків протоколів TCP/IP, IPX/SPX,SS7 (рівні MTP, SCCP, MAP, ISUP), кореляція SS7 з моделлю OSI; знання, розуміння та застосування VoIP/SIP: загальні принципи, callflow; кореляція стеку протоколів TCP/IP з моделлю OSI (призначення протоколів, адресація, маршрутизація, загальні принципи функціонування IP-мереж). Застосування протоколів маршрутизації OSPF, RIP, BGP, EIGRP в мультисервісних мережах. Особливості застосування технологій PDH, SDH, xWDM, MPLS. Протоколи IP, HTTP та DHCP для WEB – технологій. DHCPv4 та DHCPv6. Особливості застосування SMPP. Поняття мобільності в стільникових мережах зв'язку. Структури GSM мережі (включно з призначенням та функціями її елементів - HLR, MSC, VLR, SMSC, GMSC, SGSN, GGSN тощо). Структура LTE мережі (архітектура та призначення основних елементів).
Чому це цікаво/треба вивчати	Стрімкий розвиток сучасних інформаційних технологій надає нові можливості для використання на різних рівнях комп'ютерних мереж, мережевих протоколів різної функціональності для розвитку відповідних інформаційних сервісів і організації взаємодії інформаційного простору з їх партнерами та клієнтами, використання мобільного середовища та різнорідних джерел інформації і пристроїв комунікації.
Чому можна навчитися (результати навчання)	Знання та розуміння моделі OSI/ISO. Знання, розуміння та застосування стеку протоколів SS7. Знання, розуміння та застосування VoIP/SIP. Знання, розуміння та застосування стеку протоколів TCP/IP. Знання, розуміння та застосування PDH, SDH, xWDM, MPLS.

	Знання, розуміння та застосування SMPP та SIGTRAN (основні рівні, що відповідають за передачу трафіку та їх основні мітки). Знання та розуміння поняття мобільності в стільникових мережах зв'язку та яким чином вона забезпечується. Знання та розуміння структури GSM та LTE мереж.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Здатність до проектування, розробки, налагодження та вдосконалення системного, комунікаційного та програмно-апаратного забезпечення інформаційних систем та технологій, Інтернету речей (IoT), комп'ютерно-інтегрованих систем та системної мережної структури, управління ними. Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші). Здатність управляти якістю продуктів і сервісів інформаційних систем та технологій протягом їх життєвого циклу. Здатність вибору, проектування, розгортання, інтегрування, управління, адміністрування та супроводжування інформаційних систем, технологій та інфокомунікацій, сервісів та інфраструктури організації. Здатність управляти та користуватися сучасними інформаційно-комунікаційними системами та технологіями (у тому числі такими, що базуються на використанні Інтернет).
<b>Інформаційне забезпечення</b>	Силабус дисципліни, електронний конспект та презентація
<b>Форма проведення занять</b>	Лекції, комп'ютерний практикум
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	д.т.н., професор Жураковський Б.Ю.

<b>Дисципліна</b>	<b>Програмні засоби проектування та реалізації нейромережових систем</b>
<b>Кафедра</b>	Інформаційних систем та технологій
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	3 курс, 6 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: Інтелектуальні системи та технології, програмування мовою Python
<b>Що буде вивчатися</b>	Проектування, реалізація та дослідження програмних систем з використанням штучного інтелекту. Методи створення, навчання і функціонування нейромережових систем та систем глибокого навчання. Програмні засоби реалізації та дослідження нейромережових систем на Python з використанням бібліотек TensorFlow і Keras.



<b>Чому це цікаво/треба вивчати</b>	Машинне навчання – розділ інформатики, який вивчає розробку алгоритмів, які можна навчати. Одним з розділів машинного навчання є штучні нейронні мережі. Глибоке навчання є однією з найгарячіших галузей науки про дані, оскільки багато тематичних досліджень дають вражаючі результати в робототехніці, розпізнаванні зображень та штучному інтелекті. Однією з найпотужніших і простих у використанні бібліотек Python для розробки та оцінки програмних моделей глибокого навчання є Keras. Він обгортає ефективні чисельні бібліотеки обчислень Theano та TensorFlow. Вивчення даних програмних технологій реалізації та дослідження нейромережових систем дозволить легко та просто реалізовувати складні програмні системи штучного інтелекту.
<b>Чому можна навчитися (результати навчання)</b>	При вивченні даної дисципліни, студенти вивчать теоретичні основи нейронних мереж та отримають початковий досвід в розробці програмного забезпечення, що реалізує нейромережові технології. На лабораторних заняттях отримають початковий досвід створення програмних систем на основі нейронних мереж. Проведуть розробку та дослідження нейромережових систем різного призначення за допомогою мови програмування Python та бібліотек TensorFlow і Keras.
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	Набуті знання можна використати при розробці програмних систем зі штучним інтелектом.
<b>Інформаційне забезпечення</b>	Силабус, конспект лекцій, презентаційні матеріали, методичні вказівки до виконання лабораторних робіт
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., доцент Шимкович В.М.

<b>Дисципліна</b>	<b>Професійне використання SQL та PL/SQL на прикладі РСУБД Oracle 11g</b>
<b>Кафедра</b>	Інформаційних систем та технологій
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	3 курс, 6 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з теорії реляційних баз даних та проектування реляційних баз даних в розділі інфологічного та фізичного моделювання. Основи роботи в SQL, мінімальне адміністрування та налаштування серверу Oracle Database 11gXE.
<b>Що буде вивчатися</b>	В курсі буде вивчатись PL/SQL в розділах створення збережуваних процедур, функцій, тригерів, динамічний SQL та

	PL/SQL. Більш глибоко відпрацюємо запити на виборку, складні запити, аналіз та оптимізація запитів.
<b>Чому це цікаво/треба вивчати</b>	При створенні програмних застосувань важливе місце відводиться саме базі даних, написанню скриптів та запитів для коректної, швидкої роботи застосувань. PL/SQL одна з найуживаніших мов на рівні T-SQL, PL/pgSQL та інші. Знання базових конструкцій підвищить конкурентоспроможність розробника серед інших претендентів.
<b>Чому можна навчитися (результати навчання)</b>	Ознайомитись і навчитись писати складні конструкції мови PL/SQL, аналізувати запити DML, застосовувати PL/SQL для певного програмного застосування.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Набуті знання можна застосовувати при проектуванні WEB застосувань, адмініструванні реляційних серверів баз даних, програмування мовою PL/SQL.
<b>Інформаційне забезпечення</b>	Навчальна і робоча програми дисципліни, PCO, методичні вказівки з виконання лабораторних робіт (планується), конспект лекцій (планується), презентаційний матеріал та файли з прикладом скриптів (планується).
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., доцент Ульяницька К.О.

<b>Дисципліна</b>	<b>Цифрова обробка сигналів</b>
<b>Кафедра</b>	Обчислювальної техніки
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	4 курс, 7 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська, англійська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: Вища математика, Програмування, Дискретна математика, Архітектура комп'ютерів, Комп'ютерні системи
<b>Що буде вивчатися</b>	Математичні основи та алгоритми цифрової обробки сигналів (ЦОС) і зображень, а також методи їх складання, програмування та апаратної реалізації.
<b>Чому це цікаво/треба вивчати</b>	Сучасні апаратні та програмні засоби, що поширені в ІТ-галузі дуже часто ґрунтуються на виконанні алгоритмів ЦОС. Це, наприклад, засоби автоматизації, телекомунікації, обробки зображень, комп'ютерні ігри, розпізнавання образів та штучний інтелект, IoT. Без знань алгоритмів ЦОС практично неможливо тестувати, модернізувати та розробляти такі засоби.
<b>Чому можна навчитися (результати навчання)</b>	— вибирати, моделювати та модернізувати алгоритми ЦОС; — розробляти застосунки, які виконують алгоритми ЦОС.
<b>Як можна користуватися</b>	— проводити аналіз, діагностування, вибір існуючих засобів ЦОС, — виконувати модернізацію існуючих та розробку нових засобів

набутими знаннями і уміннями (компетентності)	ЦОС, — впроваджувати алгоритми ЦОС у засобах IoT та штучного інтелекту. Знати основи ЦОС та мати досвід її програмування — це обов'язкова вимога при прийомі на роботу у більшості фірм, що займаються розробкою та поширенням застосунків та апаратури для автоматизації, телекомунікації, штучного інтелекту, IoT.
Інформаційне забезпечення	Силабус дисципліни, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	д.т.н., професор Сергієнко А.М.

Дисципліна	Технології Computer Vision
Кафедра	Обчислювальної техніки
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	4 курс, 7 семестр
Обсяг	4 кредити ЄКТС, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з програмування, архітектури обчислювальних систем і мереж, дискретної математики, комп'ютерної логіки, процесів інженерії програмного забезпечення.
Що буде вивчатися	<p>Дисципліна має на меті: надання ґрунтового теоретичного базису та потужних практичних навичок програмної реалізації методів, математичних моделей і алгоритмів технологій Computer Vision (комп'ютерного бачення = комп'ютерного зору).</p> <p>Теоретичні основи Computer Vision надаються у формі лекційних занять з обов'язковою демонстрацією практичної реалізації розглянутих алгоритмів у формі прикладів програмного коду.</p> <p>Практичні навички застосування технологій Computer Vision набуваються на лабораторних заняттях, які побудовані за принципом нарощування функціональності розроблених скриптів. При цьому приділяється увага процесам інженерії програмного забезпечення.</p> <p>Практична частина дисципліни орієнтована на застосування мови програмування високого рівня Python з вивченням можливостей графічних бібліотек Graphics, Tkinter, Matplotlib, NumPy (для «сирої» реалізації алгоритмів Computer Vision) та спеціалізованих пакетів типу PIL/ Pillow, OpenGL, OpenCV для створення програмних модулів завершеної практичної спрямованості.</p> <p>Дисципліна розкриває суть етапів класичного конвеєра обробки цифрових графічних зображень: синтез зображення (просторове перетворення) – растеризація (реалістичне зображення, цифрова обробка) – векторизація (ідентифікація об'єкту та цільова обробка зображень). Зазначене реалізується у темах:</p> <p>1. Синтез цифрового зображення (2D, 3D – об'єкти, графічний метод,</p>

	<p>аналітичні та неаналітичні методи);</p> <p>2. Базові геометричні перетворення 2D, 3D об'єктів (масштабування, переміщення, обертання, проекції);</p> <p>3. Растеризація і растрові зображення, принципи створення, характеристики та алгоритми обробки (растеризація, зміна яскравості, фільтрація);</p> <p>4. Векторизація і векторні зображення, математичні моделі, алгоритми апроксимації, інтерполяції та згладжування (МНК, Сплайни, Криві Безье тощо);</p> <p>5. Моделі реалістичних зображень, доповнена реальність, моделі світла і тіні: 3D-конвеєр, рендерінг;</p> <p>6. Цифрова обробка зображень: фільтрація; корекція кольору; сегментація; векторизація; морфологічна обробка; розпізнавання.</p> <p>7. Методологічні, технологічні та прикладні аспекти ComputerVision: обробка сенсорних даних; навігація; штучний інтелект; управління.</p>
<b>Чому це цікаво/треба вивчати</b>	Опанування знань, умінь та навичок, що потребують посади: Software Developer with Computer Vision; Embedded developer for Computer Vision systems; Computer Vision Research Engineer тощо.
<b>Чому можна навчитися (результати навчання)</b>	<p>1. Знання та навички застосування і реалізації базових алгоритмів Computer Vision:</p> <ul style="list-style-type: none"> <li>- синтез 2D, 3D графічних об'єктів та їх геометричні перетворення;</li> <li>- синтезу та обробки растрових зображень;</li> <li>- синтезу та обробки векторних зображень;</li> <li>- формування реалістичних зображень;</li> <li>- цифрової обробки зображень;</li> <li>- прикладних аспектів Computer Vision;</li> <li>- комплексної реалізації етапів конвеєру цифрової обробки зображень для задач Computer Vision та інших прикладних галузей.</li> </ul> <p>2. Знання і навички застосування базових бібліотек Python для реалізації алгоритмів Computer Vision: Matplotlib, NumPy PIL/ Pillow, OpenCV.</p>
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	<p>Набуті компетенції з Computer Vision можливо застосовувати на проєктах наступних прикладних галузей і технологій:</p> <p>1. Реконструкція 3D сцени та навігація в ній (Structure-from-Motion, Road Scene Understanding and Autonomous Driving);</p> <p>2. Сегментація сцени і розуміння сюжету (Significance maps, video and 3D segmentation, 3D stream, multiple tracking, object detection, activity detection and prediction, group analysis, object detection and recognition);</p> <p>3. Розпізнавання зображень, ідентифікація об'єктів (Convolution networks, Deep learning, Image Retrieval, Object Detection).</p>
<b>Інформаційне забезпечення</b>	<p>Силабус дисципліни, навчально-методичний комплекс:</p> <p><a href="https://drive.google.com/drive/folders/1qXc3Ami_HNiVaTPIY7jLom0YRwtiGTQr?usp=sharing">https://drive.google.com/drive/folders/1qXc3Ami_HNiVaTPIY7jLom0YRwtiGTQr?usp=sharing</a><a href="https://classroom.google.com/c/NDIxODIzNTEyNDc2?cjc=mgwrbui">https://classroom.google.com/c/NDIxODIzNTEyNDc2?cjc=mgwrbui</a></p>
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	д.т.н., професор Писарчук О.О.

## Освітній компонент 9Ф-каталогу

Дисципліна	Основи WEB – технологій
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 150
Курс, семестр	4 курс, 7 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: знання мов програмування, базові знання роботи з базами даних, робота в мережі інтернет.
Що буде вивчатися	Клієнтська частина: HTML5, CSS3, JavaScript, Об'єкти в JavaScript. Події в JavaScript. DOM JS. Валідація форми. Отримання даних із зовнішніх джерел (fetch, axios). Серверна частина: Розробка серверних додатків мовою Node.js. Робота з мережевими запитами. Аутентифікація та авторизація. REST API. Робота з БД MongoDB. Протокол WebSocket. Використання Socket.io для розробки чат-додатків. WebHook, розробка TelegramBot.
Чому це цікаво/треба вивчати	Fullstack-розробники – універсальні програмісти, які розробляють і серверну, і клієнтську частину проекту. Ці фахівці дуже затребувані, тому що можуть замінити програмістів різного профілю. Fullstack-розробники володіють широким стеком технологій, тому можуть створити веб-проект поодиночі.
Чому можна навчитися (результати навчання)	Розробляти сайти статичної та динамічної структури. Розробляти інтерфейс користувача для веб-додатків. Розробляти структуру бази даних для веб-додатків. Створювати серверні програми, які обробляють інформацію, що зберігається у БД.
Як можна користуватися набутими знаннями і вміннями (компетентності)	<ul style="list-style-type: none"> <li>▪ для розробки динамічних WEB-сторінок</li> <li>▪ для розробки інтерфейсу користувача за допомогою JS.</li> <li>▪ розробляти серверні додатки мовою Node.js;</li> <li>▪ розробляти серверні додатки які обробляють інформацію, що зберігається у БД.</li> <li>▪ розробляти чат-додатки використовуючи WebSocket;</li> <li>▪ розробляти TelegramBot;</li> </ul>
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, PCO, навчально-методичний комплекс HTML5, CSS3, Node.js, MongoDB, WebSocket, socket io, telegramBot, VSCode, GitHUB.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	к.т.н., доцент Голубєв Л.П.

Дисципліна	Системи управління мережами
Кафедра	Інформаційні системи та технології

<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	4 курс, 7 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання курсів: «Комп'ютерні мережі», «Основи комп'ютерних систем та мереж», «Технології та протоколи мультисервісних мереж», «Інтернет-технології».
<b>Що буде вивчатися</b>	<ul style="list-style-type: none"> <li>• Системи та мережі управління</li> <li>• Принципи побудови систем управління сучасних мереж</li> <li>• Методи управління в мережах</li> <li>• Перспективні технології управління послугами</li> <li>• Управління трафіком та чергами на мережевих вузлах</li> <li>• Мережеві протоколи</li> <li>• Концепція TMN (Telecommunication Management Network)</li> <li>• Мережа наступного покоління (NGN)</li> <li>• Концепція IP Multimedia Subsystem</li> <li>• Softswitch</li> <li>• Концепція інтелектуальної мережі</li> <li>• Програмно-керовані мережі (SDN)</li> </ul>
<b>Чому це цікаво/треба вивчати</b>	<p>Синтез оптимальних та близьких до них систем управління складає зміст технічної кібернетики – галузі теоретичної кібернетики, яка вивчає процеси управління в живих організмах і машинах.</p> <p>Проблема управління мережами є однією з найважливіших у практиці експлуатації мереж. Загальновизнаною концепцією управління є концепція TMN, яка передбачає такі рівні управління: елементами мережі, мережею, послугами та бізнесом.</p>
<b>Чому можна навчитися (результати навчання)</b>	Якісне використання ресурсів в мережах, універсальне застосування технологій на базі мережевих протоколів, навчитися роботі в команді. Аргументувати вибір програмних та технічних засобів для створення інформаційних систем управління мережами та технологій на основі аналізу їх властивостей, призначення і технічних характеристик з урахуванням вимог до системи і експлуатаційних умов; мати навички налагодження та тестування програмних і технічних засобів інформаційних систем управління мережами
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>• Розбиратися в побудові систем управління локальних та глобальних мереж;</li> <li>• Застосовувати протоколи в системах управління мережами;</li> <li>• Застосовувати базові технології розробки мережевих протоколів на базі стандартів еталонних моделей OSI/ISO;</li> <li>• Застосовувати знання фундаментальних наук, системного аналізу, технологій моделювання та виконувати синтез систем керування компонентів нижнього рівня інформаційних систем;</li> <li>• Застосовувати знання складу, структури, принципів реалізації та функціонування інформаційно-керуючих систем та виконувати розроблення, підтримку та супроводження інформаційних та програмно-технічних засобів, оцінювати ефективність використання інформаційно-керуючих систем на підприємствах;</li> <li>• Здатність інтегрувати програмні, технічні, інформаційні та інтелектуальні компоненти усіх рівнів ієрархії інформаційно-керуючих систем в єдину розподілену систему.</li> </ul>

Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, онлайн документація та альтернативні онлайн лекції
Форма проведення занять	Лекції, лабораторні роботи (комп'ютерний практикум)
Семестровий контроль	Залік
Викладач	Д.т.н., професор Жураковський Б.Ю.

Дисципліна	Інженерна комп'ютерна графіка
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	<b>Окрім тих студентів що раніше вивчали «Комп'ютерна графіка та мультимедіа» в доцента Поліщука М.М.!!!</b> Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 6 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: Вміти користуватися комп'ютером на рівні спеціалізованого користувача, вміти працювати з растровими та векторними документами, базові знання в області комп'ютерної графіки, налагодження інтерфейсу користувача в межах стандартних функцій, наданих розробником типових програм з обробки графічних зображень.
Що буде вивчатися	Програми проектно-конструкторських редакторів AutoCAD та Spotlight Pro для автоматизованого проектування. Параметричні бази даних систем автоматизованого проектування. Програми моделювання технічних компонентів в галузі комп'ютеризованих систем. Операції налагодження графічного середовища AutoCAD. Процедури виконання різноманітних схем та креслень інженерної комп'ютерної графіки. Операції налагодження периферійного обладнання для вводу та виводу графічної інформації.
Чому це цікаво/треба вивчати	Цей курс має на меті познайомити слухача з тим, як працюють програми інженерної комп'ютерної графіки. Прикладом таких програм можуть бути 2D та 3D графічні зображення в сучасних інженерних проектах. Комп'ютерна графіка це область інформаційних технологій (ІТ), в якій використовуються, в т.ч. і засоби САПР – систем автоматизованого проектування, без яких не можливо існування сучасного промислового виробництва, а значить і успішна кар'єра майбутнього фахівця в галузі інформаційних технологій.
Чому можна навчитися (результати навчання)	<b>Знання:</b> – програм проектно-конструкторських редакторів AutoCAD, Spotlight Pro для автоматизованого проектування; – параметричних баз даних систем автоматизованого проектування; – складу основного та периферійного обладнання АРМ – автоматизованого робочого місця проектувальника. <b>Вміння:</b> – виконувати налагодження графічного середовища AutoCAD;

	– виконувати різноманітні схеми та креслення. <i>Досвід:</i> – розробки фрагментів креслень функціональних модулів робототехнічних пристроїв; налагоджування периферійного обладнання для вводу та виводу графічної інформації.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Використовувати отримані знання для розробки проєктів за допомогою комп'ютерних спеціалізованих графічних пакетів, що являють собою інструменти САПР.
Інформаційне забезпечення	Силабус дисципліни, контрольні завдання, методичні рекомендації до виконання комп'ютерного практикуму. Навчальний посібник «CAD-системи та мультимедіа».
Форма проведення занять	Лекції, комп'ютерний практикум
Семестровий контроль	Залік
Викладач	к.т.н., доцент Поліщук М.М.

Дисципліна	Ігровий штучний інтелект
Кафедра	Інформатики та програмної інженерії
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	4 курс, 7 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	знання комп'ютерної дискретної математики, алгоритмів та структур даних, основ проектування алгоритмів, підходів до програмування, (наприклад, об'єктно-орієнтоване програмування), засобів візуалізації інтерфейсу користувача
Що буде вивчатися	<ul style="list-style-type: none"> <li>інтелектуальні агенти та середовища;</li> <li>інтелектуальний пошук та взаємодія;</li> <li>управління інтелектуальними агентами;</li> <li>моделювання поведінки інтелектуальних агентів;</li> <li>прийняття простих та складних рішень інтелектуальними агентами;</li> </ul> застосування нейронних мереж в ігровому штучному інтелекті.
Чому це цікаво/треба вивчати	отримані знання та набуті практичні навички дозволять виробити певне мислення до процесу розробки ігрового штучного інтелекту та його складових
Чому можна навчитися (результати навчання)	вивчити комплексний підхід до проектування та програмування штучного інтелекту у комп'ютерних іграх
Як можна користуватися набутими знаннями і уміннями (компетентності)	<ul style="list-style-type: none"> <li>програмування поведінки ігрового середовища в цілому та програмування інтелектуальних агентів (NPC, ботів, мобів) у ігровому середовищі;</li> <li>застосування алгоритмічних рішень для розробки ігрового штучного інтелекту різних рівнів складності;</li> </ul> використання нейронних мереж для адаптації комп'ютерного опонента під стратегії гравця у комп'ютерних іграх.



Інформаційне забезпечення	Навчальна та робоча навчальна програма дисципліни, РСО, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
Форма проведення занять	Лекції та лабораторні (комп'ютерні практикуми)
Семестровий контроль	Залік
Викладач	ст. викладач Головченко М.М.

Дисципліна	Створення інформаційних систем (ІС) на базі Low-code платформ
Кафедра	Інформатики та програмної інженерії
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	4 курс, 7 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Володіння навиками бізнес-моделювання Володіння навиками програмування
Що буде вивчатися	<ul style="list-style-type: none"> <li>• BPMN - моделювання</li> <li>• Технологія Low-Code систем</li> </ul> Практичні аспекти використання
Чому це цікаво/треба вивчати	Low-code — це підхід до створення, налаштування і модифікації ІС, який практично не вимагає написання програмного коду. Low-code платформи використовують візуальні інтерфейси та діаграми представлення бізнес-процесів (наприклад, у BPMN-нотації). Це дає можливість використовувати результати роботи аналітиків напряму (автоматично) без участі розробників в процесі побудови ІС. Технологія потребує написання програмного коду в мінімальному об'ємі.
Чому можна навчитися (результати навчання)	Теоретично та практично розібратись з технологіями створення ІС с написанням програмного коду в незначних об'ємах. Створювати ІС на базі застосування Low-code систем
Як можна користуватися набутими знаннями і уміннями (компетентності)	Створювати інформаційні системи, користуючись даними проведення бізнес-аналізв
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, документація (в тому числі онлайн) на системи класу Low-code
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік, 4 курс 7 семестр
Викладач	доц., к.т.н. Новінський В.П.

Дисципліна	Мобільні комп'ютерні мережі
Кафедра	Обчислювальної техніки

<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	4 курс, 7 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: Комп'ютерні мережі, Системне програмування, Архітектура комп'ютерів
<b>Що буде вивчатися</b>	Основні концепції побудови, архітектура, комунікаційна система мобільних комп'ютерних мереж, бездротові широкосмугові мережі, бездротової персональної мережі, бездротові 5G мережі, програмно-конфігуровані мобільні мережі, мобільні хмарні технології, мобільний інтернет, основи безпеки мобільних комп'ютерних мереж,
<b>Чому це цікаво/треба вивчати</b>	Курс спрямований на формування теоретичних знань та практичних навичок із побудови, керування, модернізації, моніторингу сучасних мобільних комп'ютерних мереж.
<b>Чому можна навчитися (результати навчання)</b>	Засвоїти принципи формування теоретичних знань та практичних навичок із побудови, експлуатації, моніторингу та аналізу продуктивності сучасних мобільних комп'ютерних мереж.
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	Здатність підібрати тип, планувати і реалізовувати мобільні комп'ютерні мережі, керувати мережними ресурсами, підібрати комплекс необхідних апаратно-програмних засобів для мобільної комп'ютерної мережі, розширювати і модернізувати мережі, здійснювати моніторинг та аналіз продуктивності, діагностувати та розв'язувати проблеми.
<b>Інформаційне забезпечення</b>	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	д.т.н., професор Кулаков Ю.О.

<b>Дисципліна</b>	<b>Основи комп'ютерного моделювання</b>
<b>Кафедра</b>	Обчислювальної техніки
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	4 курс, 7 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: Програмування, Об'єктно-орієнтоване програмування, Алгоритми та методи обчислень, Системне програмування, Алгоритми і структури даних, Вступ до операційної системи Linux, Дискретна математика, Інженерія програмного забезпечення, Теорія ймовірності
<b>Що буде вивчатися</b>	- Імітаційні, математичні, логічні, еволюційні, структурно-лінгвістичні методи моделювання. Основи аналогового та фізичного

	<p>моделювання. Методологія моделювання, як технології наукових досліджень та проектування комп'ютерних систем та їх програмного забезпечення. - Методи створення моделей з урахуванням вимог до їх адекватності, способи та прийоми їх програмної реалізації та використання на практиці, методики оцінки отриманих за допомогою моделей результатів. Використання моделей для оптимізації розробки та оцінки її надійності. Методи оцінки адекватності моделей.</p> <p>- Найбільш поширені на практиці моделі комп'ютерних систем – мережі Петрі та програмні пакети моделювання. - Математичні моделі оптимізації процесів та прийоми їх програмної реалізації. Динамічне та цілочисельне програмування. Методи еволюційного моделювання на основі генетичних алгоритмів та технологій штучного інтелекту.</p>
<b>Чому це цікаво/треба вивчати</b>	<ul style="list-style-type: none"> <li>- отриманні знання та практичні навички дозволяють науково та методично грамотно застосувати моделювання для вирішення широкого спектру практичних задач проектування систем, програмного забезпечення, організації передачі даних в локальних та глобальних мережах, оцінки надійності програм та рівня інформаційної безпеки.</li> <li>- віднаходити за допомогою моделювання оптимальні рішення проектних рішень та природничо-наукових та економічних проблем.</li> <li>- проводити аналіз поведінки систем та програм в різних, в тому числі критичних ситуаціях, а також при наявності помилок при їх розробці.</li> </ul>
<b>Чому можна навчитися (результати навчання)</b>	<ul style="list-style-type: none"> <li>- Аналізувати можливості вирішення широкого кола практичних задач за допомогою методів моделювання; здійснювати обґрунтований вибір моделі та її побудову з урахуванням вимог до її адекватності, грамотно оцінювати надійність отриманих результатів моделювання;</li> <li>- оволодіти практичними способами та прийомами програмування імітаційних, математичних та еволюційних моделей</li> <li>- Використовувати математичні моделі оптимізації процесів та динамічного програмування для створення ефективних програм.</li> <li>- Оцінювати з використанням комп'ютерних моделей параметри якості систем та програм, зокрема надійності їх роботи в різних ситуаціях, а також рівень безпеки даних та програм</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>- Використовувати арсенал методів комп'ютерного моделювання в якості інструменту наукових досліджень в технічних, природничо-наукових, медичнських, економічних та соціальних сферах</li> <li>- Застосовувати методи комп'ютерного моделювання та існуючі програмні продукти для ефективного проектування систем та програм з урахуванням заданих вимог до їх якісних характеристик</li> <li>- Здійснювати оптимізацію проектних рішень з застосуванням комп'ютерного моделювання;</li> <li>- Проводити аналіз поведінки систем та програм в критичних ситуаціях, оцінювати їх надійність, виконувати пошук помилок.</li> </ul>
<b>Інформаційне забезпечення</b>	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., доцент Волокита А.М., асистент Іваніщев Б.В

Дисципліна	Технології Data Science
Кафедра	Обчислювальної техніки
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	<b>Окрім студентів що раніше вивчали «Вступ до Технології Data Science»!!!</b> Кількість студентів на курс від 60 до 150
Курс, семестр	4 курс, 7 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з програмування, архітектури обчислювальних систем і мереж, дискретної математики, комп'ютерної логіки, процесів інженерії програмного забезпечення.
Що буде вивчатися	<p>Дисципліна має на меті: надання комплексного ґрунтовного теоретичного базису та потужних практичних навичок програмної реалізації методів, математичних моделей і алгоритмів технологічних процесів Data Science (дослідження даних): дані – інформація – знання – маніпулювання знаннями - візуалізація.</p> <p>Специфіка курсу полягає у розгляді, поряд із класичними методологіями Data Science, передових авторських розробок, отриманих у ході реалізації практичних R&amp;D проєктів.</p> <p>Теоретичні основи Data Science надаються у формі лекційних занять з обов'язковою демонстрацією розглянутих алгоритмів у формі прикладів програмного коду.</p> <p>Практичні навички застосування технологій Data Science набуваються на лабораторних заняттях. При цьому приділяється увага процесам інженерії програмного забезпечення.</p> <p>Практична частина дисципліни орієнтована на застосування мови програмування високого рівня Python з вивченням функціоналу бібліотек: Numpy, Pandas, Scipy, Scikit-learn, Tensorflow, Keras, Requests, OpenCV, Pillow, Matplotlib.</p> <p>В дисципліні розкривається суть технологічних процесів Data Science: обробка даних з метою отримання інформації – обробка інформації з метою виявлення знань – використання навичок на практиці - візуалізація результатів.</p> <p>Дисципліна складається із двох взаємопов'язаних блоків:</p> <p>1. Методологічні основи Data Science:</p> <p>1.1. Прикладний статистичний аналіз даних – статистичне навчання (моделі даних; статистичний аналіз характеристик експериментальної вибірки; обробка аномальних вимірів; оцінювання, екстраполяція та інтерполяція трендовими моделями - рекурентне згладжування і згладжування накопиченої вибірки; побудова нелінійних моделей експериментальних даних з використанням диференціальних перетворень);</p> <p>1.2. Багатокритеріальні методи прийняття рішень (багатокритеріальне оцінювання; багатокритеріальна ідентифікація; багатокритеріальний розподіл ресурсів; багатокритеріальний структурно-параметричний синтез систем);</p> <p>1.4. Інтелектуальний аналіз даних (технології: OLAP, Data Mining, Text Mining, Image Mining, Knowledge discovery, Speech and language</p>

	<p>recognition);</p> <p>1.3. Застосування штучного інтелекту для технологій Data Science (штучні нейронні мережі; методи моделі і алгоритми самоорганізації та ситуативного аналізу).</p> <p>2. Технологічні аспекти Data Science:</p> <p>2.1. Алгоритми та технології прогнозування динаміки змін фінансових та фондових ринків (за статистичними і альтернативними моделями);</p> <p>2.2. Алгоритми та технології визначення кредитних ризиків для банківських CRM систем (скорингові та багатокритеріальні моделі);</p> <p>2.3. Алгоритми ідентифікації поточних ситуацій для виробничих CRM систем і об'єктах критичної інфраструктури (технології багатофакторного аналізу та Computer Vision).</p>
<b>Чому це цікаво/треба вивчати</b>	Опанування знань, умінь та навичок, що потребують посади: DataScientist, DataEngineer; DataAnalyst – RiskTeam тощо.
<b>Чому можна навчитися (результати навчання)</b>	<p>1. Знання та навички застосування і реалізації базових алгоритмів Data Science – машинного навчання: прикладний статистичний аналіз даних; багатокритеріальні методи прийняття рішень; інтелектуальний аналіз даних; застосування штучного інтелекту для технологій Data Science; візуалізація результатів Data Science.</p> <p>1. 2. Знання і навички застосування базових бібліотек Python для реалізації алгоритмів Data Science: Numpy, Pandas, Scipy, Scikit-learn, Tensorflow, Keras, Requests, OpenCV, Pillow, Matplotlib.</p>
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	<p>Набуті компетенції з Data Science можливо застосовувати на проектах наступних прикладних галузей і технологій:</p> <p>1. Аналіз даних для задач електронної комерції;</p> <p>2. Аналіз даних для промислових та інфраструктурних CRM систем;</p> <p>3. Аналіз візуальних та геопросторових даних різного спрямування;</p> <p>4. Аналіз даних для медичних інформаційних систем.</p>
<b>Інформаційне забезпечення</b>	<p>Силабус дисципліни, навчально-методичний комплекс:  <a href="https://drive.google.com/drive/folders/1xqv2CMJ_BJmOL8QK2GjUne6c oi0T5I2U?usp=sharing">https://drive.google.com/drive/folders/1xqv2CMJ_BJmOL8QK2GjUne6c oi0T5I2U?usp=sharing</a>  <a href="https://classroom.google.com/c/NDIxODIzNTEyNDUy?cjc=qnsxkzd">https://classroom.google.com/c/NDIxODIzNTEyNDUy?cjc=qnsxkzd</a></p>
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	д.т.н., професор Писарчук О.О.

## Освітній компонент 10Ф-каталогу

Дисципліна	Імітаційне моделювання
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	4 курс, 7 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін «Програмування», «Теорія ймовірностей та математична статистика», «Емпіричні методи програмної інженерії».
Що буде вивчатися	Методи імітаційного моделювання, імітаційні моделі, програмні пакети імітаційного моделювання.
Чому це цікаво/треба вивчати	Коли явища в системі занадто складні й різноманітні, аналітична модель стає лише грубим наближенням. У таких умовах дослідник змушений застосовувати імітаційне моделювання. В імітаційній моделі поведінка системи описується за допомогою набору алгоритмів із застосуванням певної мови програмування, які реалізують ситуації, що виникають у реальній системі. Імітаційні моделі характеризуються точним відображенням певного процесу або явища. Тому математичні моделі, які входять до їх складу, виявляються досить складними, у них присутні нелінійні й стохастичні залежності й змінні.
Чому можна навчитися (результати навчання)	В результаті вивчення навчальної дисципліни студент повинен знати: <ul style="list-style-type: none"> <li>- основні види імітаційних моделей, принципи імітаційного моделювання, визначення та характеристики базової випадкової величини, приклади базових датчиків, основні методи імітації випадкових величин з дискретними та абсолютно неперервними розподілами, принципи моделювання випадкових векторів та процесів, зокрема, систем масового обслуговування;</li> <li>- методи верифікації результатів моделювання.</li> </ul> вміти: <ul style="list-style-type: none"> <li>- моделювати значення базової випадкової величини, проводити верифікацію отриманих результатів моделювання, моделювати вибірки заданих об'ємів для випадкових величин з дискретними та абсолютно неперервними розподілами, моделювати значення випадкових векторів та процесів;</li> <li>- застосовувати імітаційні моделі для прогнозу та узгодженості з реальними статистичними даними.</li> </ul>
Як можна користуватися набутими знаннями і уміннями (компетентності)	Працювати з основними імітаційними моделями та застосувати отримані знання до вирішення прикладних задач.
Інформаційне забезпечення	Силабус дисципліни, опорний конспект лекцій, методичні вказівки до виконання лабораторних робіт, перелік завдань для самостійної роботи студента, перелік питань до МКР та семестрового контролю.
Форма проведення занять	Лекції, лабораторні роботи.

Семестровий контроль	Залік
Викладач	к.т.н., доцент Сокульський О.Є.

Дисципліна	Протоколи і алгоритми електронного голосування
Рівень ВО	Перший (бакалаврський)
Кафедра	Інформаційних систем та технологій
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	4 курс, 7 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: алгебра, дискретна математика, теорія інформації та кодування, комп'ютерні мережі, безпека інформаційних систем.
Що буде вивчатися	Протоколи і алгоритми електронного голосування для забезпечення надійного, достовірного (виключає фальшування), безпечного і транспарентного для всіх сторін і учасників (забезпечує достовірну перевірку), проведення і фіксації волевиявлення на виборах будь-якого рівня (від виборів старости студентської групи і представників у органи студентського самоврядування, до виборів народних депутатів і президента країни). Міжнародний досвід та вітчизняні розробки.
Чому це цікаво/треба вивчати	Будь-які вибори, де приймають участь три зацікавлені сторони: виборці, кандидати, третя (довірена?) сторона – організатор виборів, мають нарізно спрямовані вектори своїх інтересів, які перетинаються лише у одній точці – провести вибори. Один із можливих сценаріїв: виборці і кандидати мають прийняти обраним кандидата, оголошеного організатором виборів, але об'єктивно вони зацікавлені у різному: виборці воліють прийняти обраним того кандидата, за якого віддана більшість голосів, а кожен кандидат воліє, щоб його голосили обраним саме його. Організатор виборів більш за все зацікавлений оголосити вибори такими, що відбулися, щоб унеможливити претензії до себе. Досвід останніх десятиліть демонструє і більш складні сценарії і зацікавленості сторін. Як забезпечити справедливе, чесне голосування? Справедливий, неупереджений підрахунок голосів? Уникнути ефектів соціальної інженерії, корупційних чи просто агресивних впливів на виборців, кандидатів і організатора виборів? Шукаємо разом!
Чому можна навчитися (результати навчання)	Створенню, аналізу, обґрунтуванню вибору Протоколів і алгоритмів електронного голосування для забезпечення надійного, достовірного (виключає фальшування), безпечного і транспарентного для всіх сторін і учасників (забезпечує достовірну перевірку), проведення і фіксації волевиявлення на виборах будь-якого рівня (від виборів старости студентської групи і представників у органи студентського самоврядування, до виборів народних депутатів і президента країни). Шляхи створення і налаштування відповідного ПЗ.
Як можна користуватися набутими знаннями і вміннями	Створювати, аналізувати, обґрунтовувати вибір та застосовувати Протоколи і алгоритми електронного голосування для забезпечення надійного, достовірного (виключає фальшування).

(компетентності)	вання), безпечного і транспарентного для всіх сторін іучасників (забезпечує достовірну перевірку), проведення іфіксації волевиявлення на виборах будь-якого рівня (від виборівстаростистудентськоїгрупиіпредставниківіуорганістудентсько го самоврядування, до виборів народних депутатів іпрезидента країни). Створювати і налаштувати відповідне ПЗ.
Інформаційне забезпечення	Силабус, конспект лекцій, перелік питань і завдань до семестрового контролю, навчальний посібник до виконаннялабораторних робіт, завдання до виконання лабораторних робіт, завдання до самостійної роботи студентів та методичні вказівкидо їх виконання.
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	к.т.н., доцент Полтораєк В.П.

Дисципліна	Технології інтернет речей
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	<b><u>Окрім студентів груп ІК, що навчаються за ОП - Інформаційне забезпечення робототехнічних систем</u></b> Кількість студентів на курс від 60 до 150
Курс, семестр	4 курс, 7 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: фізика, системотехніка, програмування, комп'ютерні мережі
Що буде вивчатися	Екосистема Інтернету речей. Архітектура Інтернету Речей (Рівень датчиків, Мережевий рівень, Рівень обробки даних, Рівень додатків). Еталонні моделі IoT. IoT платформи (платформа Linux Foundation, платформа AggreGate, платформа Everyware Cloud). Моделі IoT шлюзів. Прості та інтелектуальні сенсори. Сенсорно-комп'ютерні системи. Безпроводна передача та обробка даних Інтернет речей. Технології та протоколи передачі даних на довгі відстані в IoT мережах: LoRaWAN, SigFox, NB-IoT, Weightless-P. Технології та протоколи передачі даних на короткі відстані в IoT мережах: Z-Wave, NFC, RFID, Bluetooth Low Energy, Wi-Fi HaLow. Сенсорні мережі. Смарт грід на рівні квартири або будинку. Системи на базі технологічної платформи Смарт грід. Smart City. Платформи Smart City. Smart Home. Smart Factory. Smart Manufacturing.
Чому це цікаво/треба вивчати	Смарт-середовище - це місце, в якому традиційні системи працюють більш ефективно за рахунок використання інформаційно-комунікаційних технологій. Інформаційно-комунікаційні технології дозволяють використовувати менше енергетичних ресурсів, задовольняючи незмінний обсяг потреб, та зменшувати масштаби парникової емісії. Це означає запровадження «розумнішої» системи міського транспорту, оновленої системи водопостачання та утилізації відходів, а також створення ефективніших систем опалення та



	охолодження будинків тощо.
<b>Чому можна навчитися (результати навчання)</b>	Демонструвати знання сучасного рівня технологій інформаційних систем, практичні навички програмування та використання прикладних і спеціалізованих комп'ютерних систем та середовищ метою їх запровадження у професійній діяльності. Розуміти знання технології, принципів організації та функціонування IoT, вміти проектувати IoT-системи, демонструвати знання мікроконтролерів, платформ аштування, програмування кінцевих пристроїв, створювати вбудоване та серверне програмне забезпечення. Використовувати моделі та методи розподілу функціоналу між пристроями та взаємодії пристроя та середовища. Уміти перетворювати звичайні технічні вироби в інтернет речей, реалізовувати взаємодію інтернет речей між собою та середовищем.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Здатність до проектування, розробки, налагодження та вдосконалення системного, комунікаційного та програмно-апаратного забезпечення інформаційних систем та технологій, Інтернету речей (IoT), комп'ютерно-інтегрованих систем та системної мережної структури, управління ними. Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).
<b>Інформаційне забезпечення</b>	Силабус дисципліни, електронний конспект та лекційні матеріали
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	залік
<b>Викладач</b>	д.т.н., проф. Жураковський Богдан Юрійович

<b>Дисципліна</b>	<b>Реактивне програмування</b>
<b>Кафедра</b>	Інформатики та програмної інженерії
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	4 курс, 7 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з об'єктно-орієнтованого програмування, паралельного програмування, багатопоточності, та функціонального програмування
<b>Що буде вивчатися</b>	<ul style="list-style-type: none"> <li>• Елементи сучасних мов програмування для підтримки функціонального програмування, зокрема лямбда-виразів</li> <li>• Елементи сучасних мов програмування для реалізації асинхронного програмування</li> <li>• Визначення та базові концепції реактивного програмування</li> <li>• Елементи сучасних мов програмування для реалізації асинхронного програмування</li> </ul> Використання реактивного програмування для побудови серверних систем

<b>Чому це цікаво/треба вивчати</b>	Реактивне програмування – одна із концепцій, що зараз активно розвиваються із реалізацією у всіх провідних мовах програмування
<b>Чому можна навчитися (результати навчання)</b>	Визначати компоненти програмних систем, де можна ефективно використовувати асинхронне та реактивне програмування. Використовувати сучасні підходи функціонального, асинхронного та реактивного програмування для розробки програмного забезпечення.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Отримані знання та вміння дозволять використовувати сучасні засоби асинхронного та реактивного програмування для розробки високонавантажених ефективних серверних систем
<b>Інформаційне забезпечення</b>	Силабус дисципліни, контрольні завдання, конспект, презентації для лекцій
<b>Форма проведення занять</b>	Лекції та лабораторні (комп'ютерні практикуми)
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	доц.,к.т.н. Полупан Ю.В.

Дисципліна	Побудова IP-мереж на базі обладнання Juniper
<b>Кафедра</b>	Інформатики та програмної інженерії
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	4 курс, 7 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: комп'ютерні мережі, інтернет-технології
<b>Що буде вивчатися</b>	Архітектура пристроїв Juniper та операційної системи Junos. Командна стрічка операційної системи Junos. Базове конфігурування та технічне обслуговування пристроїв Juniper. Використання пристроїв Juniper для побудови маршрутизованих мереж. Статична маршрутизація. Налаштування статичної маршрутизації на маршрутизаторах Juniper. Протоколи маршрутизації OSPF та BGP. Налаштування протоколів маршрутизації OSPF та BGP на маршрутизаторах Juniper. Керування розповсюдженням маршрутної інформації. Політики маршрутизації та маршрутизація на основі фільтрів. Фільтрація трафіку з метою забезпечення безпеки мережі. Налаштування GRE та інших тунелів. Забезпечення якості обслуговування. Забезпечення надлишковості та відмовостійкості маршрутизованих мереж на базі обладнання Juniper. Використання комутаторів Juniper. Віртуальні локальні мережі (VLAN). Маршрутизація між VLAN. Забезпечення безпеки мереж Ethernet на базі комутаторів Juniper. Забезпечення надлишковості та відмовостійкості мереж Ethernet на базі комутаторів Juniper. STP, LAG, RTG.
<b>Чому це цікаво/треба вивчати</b>	Juniper Networks – один із лідерів з виробництва мережевого

	<p>обладнання. Juniper з кожним роком застосовується дедалі більше. При цьому система команд Juniper не схожа на систему команд інших виробників. Це відкриває нові можливості, але ускладнює вивчення обладнання.</p> <p>Знання Juniper – це конкурентна перевага при працевлаштуванні, в тому числі за кордоном. Крім того це цікаво.</p> <p>Під час навчання використовується реальна продукція Juniper Networks. Надаються навчальні матеріали.</p> <p>Ви зможете вивчити обладнання Juniper та підготуватися до сертифікації JNCIA-JUNOS та JNCIS-ENT.</p>
<b>Чому можна навчитися (результати навчання)</b>	<p>Проектувати комп'ютерні мережі на базі обладнання Juniper.</p> <p>Конфігурувати маршрутизатори та комутатори Juniper в корпоративних мережах.</p> <p>Підготуватися до сертифікаційних іспитів JNCIA-Junos та JNCIS-ENT</p>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<p>Проектувати маршрутизовані та комутовані мережі на базі обладнання Juniper.</p> <p>Виконувати базові налаштування, моніторинг та технічне обслуговування обладнання Juniper.</p> <p>Налаштовувати статичну маршрутизацію на маршрутизаторах Juniper, в т.ч. агреговані та генеровані маршрути.</p> <p>Керувати розповсюдженням маршрутної інформації та налаштовувати балансуювання навантаження.</p> <p>Розуміти логіку роботи протоколів маршрутизації OSPF, BGP та налаштовувати OSPF та BGP на маршрутизаторах Juniper.</p> <p>Розуміти концепцію тунелювання трафіку. Налаштовувати GRE та IP-IP тунелі.</p> <p>Розуміти та налаштовувати різні можливості відмовостійкості (high availability)</p> <p>Використовувати комутатори Juniper в локальних мережах.</p> <p>Конфігурувати віртуальні локальні мережі (VLAN) та маршрутизацію між VLAN на базі обладнання Juniper.</p> <p>Налаштувати засоби безпеки в локальних мережах. Налаштувати фільтрацію трафіку на пристроях Juniper.</p>
<b>Інформаційне забезпечення</b>	Силабус дисципліни, електронний конспект та презентація
<b>Форма проведення занять</b>	Лекції, практичні роботи з обладнанням Juniper
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., Миронов Дмитро Вікторович JNCIA, JNCIS, CCAI, CCNA, CCNP

<b>Дисципліна</b>	<b>Системне програмування в середовищі Unix</b>
<b>Кафедра</b>	Обчислювальної техніки
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	4 курс, 7 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська

<b>Вимоги до початку вивчення</b>	Вміння програмувати на С або С++ та вміння працювати в Unix-подібній операційній системі на рівні користувача. Базові знання з дисциплін Програмування, Структури даних та алгоритми.
<b>Що буде вивчатися</b>	Вивчатиметься системне програмування в середовищі Unix, тобто розроблення системних програм для Unix-подібних операційних систем на рівні використання системних викликів для взаємодії з ядром операційної системи. Надається ґрунтовна інформація про функції POSIX (та достатня інформація про реалізації відповідних системних викликів), які використовуються в системних програмах. Дисципліна не орієнтована на системне програмування в якійсь конкретній реалізації Unix-подібної операційної системи, вивчатиметься переносне системне програмування. Дисципліна складається з таких тем: середовище виконання програми, робота з файлами, сигнали, управління процесами, неіменовані канали, мультиплексування введення-виведення, управління адресним простором процесу, асинхронні дії та інших.
<b>Чому це цікаво/треба вивчати</b>	Цю дисципліну доцільно вивчати тим, хто розроблятиме системні програми для Unix-подібних операційних систем на рівні використання системних викликів. Програмування завдань виконується мовою програмування С або С++, але набуті знання будуть корисні для вирішення деяких системних задач для Unix-подібних операційних систем іншими мовами програмування.
<b>Чому можна навчитися (результати навчання)</b>	Розробляти системні програми для Unix-подібних операційних систем на рівні використання системних викликів мовою програмування С або С++, які працюють із файлами, працюють із сигналами, управляють процесами, працюють із неіменованими каналами, використовують мультиплексування введення-виведення, управляють адресним простором процесу, використовують асинхронні дії.
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	Набуті знання можна використовувати під час розроблення системних програм для Unix-подібних операційних систем, для супроводження вихідного коду наявних системних програм для Unix-подібних операційних систем, під час розроблення ефективніших прикладних програм.
<b>Інформаційне забезпечення</b>	Силабус, РСО, конспект лекцій, завдання на лабораторні роботи
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	Ст. викладач Сімоненко Андрій Валерійович

<b>Дисципліна</b>	<b>Основи науки про дані</b>
<b>Кафедра</b>	Обчислювальної техніки
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 150
<b>Курс, семестр</b>	4 курс, 8 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська

<b>Вимоги до початку вивчення</b>	Базові знання з вищої математики (математичний аналіз, лінійна алгебра та аналітична геометрія), теорія ймовірностей та математична статистика, програмування
<b>Що буде вивчатися</b>	Загальні поняття про сутність аналізу даних, застосування принципу Бонферроні, методу пошуку найближчих сусідів, шинглінгу документів. Принципи збереження схожості скорочених наборів, локально-чутливе хешування та його застосування, міри відстані. Поняття про аналіз потоків даних різної природи та складності та аналіз посилань. Модель ринкової корзини та алгоритм A-Priori
<b>Чому це цікаво/треба вивчати</b>	Наука про дані має вагомe практичне значення для вирішення актуальних повсякденних проблем. Застосування науки про дані змінює спосіб життя, праці й мислення, оскільки однією з умов успішного розвитку світової економіки на сучасному етапі є можливість фіксувати й аналізувати накопичені набори даних. Сучасні ефективні методи роботи з даними сприяють як підвищенню конкурентоспроможності різних сфер економічної діяльності, так і фахівців, які володіють цими методами.
<b>Чому можна навчитися (результати навчання)</b>	Створювати моделі для даних з застосуванням узагальнення та виділення ознак. Використовувати методи обчислення жакартової подібності множин. Застосовувати міри відстані для визначення подібності, зокрема, косинусну відстань, відстань редагування, відстань Хеммінга, локально-чутливі функції. Аналізувати потоки даних шляхом побудови моделі поточкових даних та фільтрації потоків, здійснювати підрахунок різних елементів у потоці, зокрема, підрахунок у вікні та згасаючих вікнах, використовувати алгоритми DGIM. Здійснювати аналіз посилань з застосуванням PageRank, його ефективне обчислення, обчислювати тематично чутливий PageRank, визначати спам посилань.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>- для формування початкових наборів даних з застосуванням принципу Бонферроні;</li> <li>- для пошуку текстуально подібних документів (плагіату, дзеркальних сторінок, статей з одного джерела);</li> <li>- для визначення подібності он-лайнних покупок та рейтингу фільмів;</li> <li>- для розпізнавання об'єктів, співставлення відбитків пальців та газетних статей;</li> <li>- для створення програмних систем ефективної фільтрації небажаних елементів потоку (вхідні потоки електронної пошти);</li> <li>- для створення телепорт-наборів.</li> </ul>
<b>Інформаційне забезпечення</b>	Силабус дисципліни, навчально-методичний комплекс
<b>Форма проведення занять</b>	Лекції, лабораторні роботи, самостійна робота
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	д.т.н., професор Новотарський М.А.

<b>Дисципліна</b>	<b>Тестування та контроль якості (QA) вбудованих систем (Сертифікатна програма. авторський курс компанії Global Logic)</b>
<b>Кафедра</b>	Обчислювальної техніки
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	4 курс, 7 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: ОС Linux, Git. Програмування, Алгоритми і структури даних, Системне програмування, Операційні системи. Комп'ютерні мережі, Архітектура комп'ютерів, Архітектура процесорів, Мікропроцесорні системи.
<b>Що буде вивчатися</b>	<p>Основи тестування. Цілі тестування на різних стадіях життєвого циклу розроблення продукту. Моделі розроблення ПЗ. Основні рівні тестування та їх взаємозв'язки. Складання тестової документації.</p> <p>Робота з обладнанням на базі платформи BeagleBone Black. Самостійне створення вбудованої системи, налаштувати оточення. Досвід тестування вбудованої операційної системи з урахуванням налаштування тестового оточення, в рамках чого розглядаються питання:</p> <ul style="list-style-type: none"> <li>- усунення несправностей у мережі (Network Troubleshooting);</li> <li>- налаштування операційної систему Linux та комп'ютерної мережі;</li> <li>- розвертання операційної системи (Linux kernel, U-Boot, BusyBox) для архітектури процесорів ARM Cortex A8 шляхом компіляції вихідного коду; різні способи прошивання мікросхеми процесора на платі;</li> <li>- виконувати тестування власного продукту на базі платформи BeagleBone Black;</li> <li>- використання Git та GitHub для контролю версій.</li> </ul> <p>Перевірка функціональних та нефункціональних атрибутів програмного та апаратного забезпечення у самостійно створеній вбудованій системі. Створення тестів для перевірки та підтвердження вбудованого програмного і апаратного забезпечення відповідно до вимог клієнта.</p>
<b>Чому це цікаво/треба вивчати</b>	<p>В ІТ компаніях велика увага приділяється якості продукту, що випускається на ринок.</p> <p>Основні причини необхідності тестування, які обґрунтовують спеціальну підготовку спеціалістів-тестувальників:</p> <ul style="list-style-type: none"> <li>- знаходження помилок в програмному та апаратному забезпеченні,</li> <li>- зниження ризику як для користувачів, так і для компанії, скорочення витрат на розробку та обслуговування,</li> <li>- підвищення продуктивності.</li> </ul> <p>Тестування убудованого програмного та апаратного забезпечення - це відмінний підхід для гарантування безпеки в таких критичних додатках, як медичне обладнання, залізниця, авіація, транспортна промисловість тощо. Великі ІТ компанії, що займаються такими розробками, запрошують на роботу спеціально підготовлених</p>

	<p>фахівців-тестувальників. Для підготовки фахівців для тестування убудованих систем необхідно базові знання в області проектування та експлуатації убудованих систем, комп'ютерних мереж.</p> <p>Курс «Тестування та контроль якості (QA) вбудованих систем» може бути прослуханий студентами як окремий курс, так і у складі <b>сертифікатної програми «Інженерія вбудованих систем та Інтернет речей»</b>, яка запроваджена на факультеті інформатики та обчислювальної техніки з 2021 – 2022 навчального року. Детально з сертифікатною програмою можна ознайомитися на сайті кафедри обчислювальної техніки <a href="https://comsys.kpi.ua/sertifikatni-programi">https://comsys.kpi.ua/sertifikatni-programi</a>.</p> <p>Перелік дисциплін сертифікатної програми:</p> <ul style="list-style-type: none"> <li>• Технології програмування C/Embedded – 5 семестр;</li> <li>• Управління IT-інфраструктурними проектами – 6 семестр;</li> <li>• Технології програмування на ПЛІС (FPGA) – 6 семестр;</li> <li>• Тестування та контроль якості (QA) вбудованих систем – 7 семестр;</li> <li>• Технології розроблення вбудованих систем IoT – 8 семестр.</li> </ul> <p>Курс викладає та є його автором сертифікований QA Engineer та фахівець компанії GlobalLogic. В цьому курсі і в сертифікатній програмі загалом розглядаються цікаві задачі розроблені у співробітництві з менторами та тренерами компанії GlobalLogic, які також приймають участь в лекційних та практичних заняттях. Ми намагаємося допомогти студентам краще зрозуміти реалії сучасного ринку Embedded Systems та потреби роботодавців, подолати важкий поріг входження в Embedded та світ HW.</p> <p>Дефіцит фахівців о області Embedded Systems та IoT на сьогодні доволі значний, тому опанування дисциплін сертифікатної програми дозволить розширити можливості студентів в пошуку майбутньої роботи та підвищить його конкурентоспроможність. З точки зору компанії GlobalLogic участь в сертифікатній програмі дозволить сформувати необхідні знання та вміння, що задовольнять її, як роботодавця.</p>
<p><b>Чому можна навчитися (результати навчання)</b></p>	<ul style="list-style-type: none"> <li>- Налаштовувати операційну систему Linux та комп'ютерну мережу;</li> <li>- Розвертати операційну систему (Linux kernel, U-Boot, BusyBox) для архітектури процесорів ARM Cortex A8 шляхом компіляції вихідного коду.</li> <li>- Використовувати різні способи прошивання мікросхеми процесора на платі. Користуватися різними інтерфейсами, а також програми емуляції терміналів для комунікації і налагодження убудованих пристроїв.</li> <li>- Виконувати тестування власного продукту на базі платформи BeagleBone Black.</li> <li>- Знаходити помилки в програмному та апаратному забезпеченні убудованих систем.</li> <li>- Усувати несправності в роботі комп'ютерної мережі.</li> <li>- Складати тестову документацію.</li> <li>- Працювати в команді.</li> </ul> <p>Отримати базову підготовку до сертифікації QA спеціаліста по ISTQB</p>
<p><b>Як можна користуватися набутими знаннями і вміннями</b></p>	<p>Отримані знання достатні для того, щоб розробити архітектуру та реалізувати власний вбудований пристрій, налагодити його та налаштувати для роботи як автономно так і в інфраструктурі IoT.</p>

<b>(компетентності)</b>	<p>Працювати в ІТ тестувальником програмного та апаратного забезпечення.</p> <p>Продовжувати підготовку в області розробника програмного та апаратного забезпечення убудованих систем.</p> <p>Продовжити підготовку в області фахівця по автоматизації тестування програмного і апаратного забезпечення.</p> <p>Отримані базові знання в області основ QA тестування можуть бути використані в області тестування програмного забезпечення, а також бути основою для самостійної підготовки до сертифікації QA спеціаліста по ISTQB</p>
<b>Інформаційне забезпечення</b>	Навчальна і робоча програми дисципліни, РСО, навчально-методичний комплекс.
<b>Форма проведення занять</b>	Лекції, практичні та семінарські заняття, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	QA Engeneer GlobalLogic Таранюк В.А., д.т.н., професор кафедри ОТ Клименко І.А.,



## Освітній компонент ІІФ-каталогу

Дисципліна	Проектування мікропроцесорних систем на базі Arduino та Raspberry Pi
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	4 курс, 7 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: Базові знання мов програмування, базові знання електротехніки, роботи в мережі інтернет.
Що буде вивчатися	Проектування мікропроцесорних систем на базі <i>Arduino</i> : Цифрові порти введення-виведення, широко-імпульсна модуляція (ШІМ), робота з аналоговими датчиками, робота з серво та кроковим двигунами, робота з LCD - дисплеєм. Проектування мікропроцесорних систем на базі міні-комп'ютерів <i>RaspberryPi</i> .
Чому це цікаво/треба вивчати	Мікропроцесорні системи призначені для автоматизації обробки інформації та управління різними процесами. Застосування мікропроцесорних систем практично у всіх електричних пристроях - найважливіша риса технічної інфраструктури сучасного суспільства. МПС на основі МК використовуються найчастіше як вбудовані системи для вирішення завдань управління деяким об'єктом.
Чому можна навчитися (результати навчання)	<ul style="list-style-type: none"> <li>▪ Проектувати мікропроцесорні системи на базі Arduino;</li> <li>▪ Працювати з цифровими портами введення-виведення;</li> <li>▪ Працювати з аналоговими датчиками та серво і кроковим двигунами;</li> </ul> Проектувати мікропроцесорні системи на базі RaspberryPi;
Як можна користуватися набутими знаннями і вміннями (компетентності)	При проектуванні мікропроцесорних систем на базі Arduino. При розробці систем автоматизації та робото-технічних систем на базі МПС. При проектуванні мікропроцесорних систем на базі RaspberryPi.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни, PCO, навчально-методичний комплекс, Arduino IDE, Tinkercad, OS Debian, Wiringpi
Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	к.т.н., доцент Голубев Л.П.

Дисципліна	Інфраструктура програмного забезпечення WEB-застосувань
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 150
Курс, семестр	4 курс, 7 семестр
Обсяг	4 кредити, 120 годин

<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: основ програмування, реляційні бази даних та основи SQL (можливе одночасне вивчення), основи HTML та CSS (можливе одночасне вивчення). Бажано основи JavaScript (не критично).
<b>Що буде вивчатися</b>	Способи аналізу функціональних вимог для сучасних програмних систем на базі мікроархітектури та їх розгортання; - створення проекту і Web-застосунків на основі типової платформи, як основу мікроархітектурного рішення; - розгортання і контейнеризація Web-застосунків на Docker. - створення мікроархітектурного рішення на основі Web-застосунків; - розгортання і контейнеризація програмної інфраструктури Web-застосунків на Docker.
<b>Чому це цікаво/треба вивчати</b>	За один семестр можна вивчити і отримати перший досвід основ Web-застосунків і розгортання програмної інфраструктури на основі Docker. Цікаво для тих, хто маючи знання основ програмування, HTML та CSS бажає отримати досвід розробки програмної інфраструктури, планує після університету працювати в цій сфері, або почати свій бізнес.
<b>Чому можна навчитися (результати навчання)</b>	Проектуванню і бізнес-аналітики основ Web-застосунку на основі типовий фреймворків. Повному циклу розробки Web-застосунку на основі мікроархітектурних рішень. Розгортанню програмної інфраструктури на основі контейнеризації, на прикладі Docker.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Отриманні знання, вміння, досвід і навички дають можливість працевлаштування у відповідній професійній сфері. Також отриманий досвід дозволяє зробити перший крок щодо створення власного бізнесу.
<b>Інформаційне забезпечення</b>	Силабус, методичні рекомендації до виконання лабораторних робіт, інформаційна підтримка у Телеграмі (група відкривається на час занять), середовище розробки, довідкова система.
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н, доцент Галушко Д.О.

<b>Дисципліна</b>	<b>Сучасні технології розробки WEB-застосувань на платформі .NET (3 частина)</b>
<b>Кафедра</b>	Інформатики та програмної інженерії
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 30 до 60
<b>Курс, семестр</b>	4 курс, 7 семестр
<b>Обсяг</b>	120 годин / 4 кредитів ЄКТС

<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	<p><u>Обов'язково:</u></p> <ol style="list-style-type: none"> <li>1. Знання синтаксису мови C#, і основних принципів роботи платформи .NET;</li> <li>2. Розуміння SOLID та OOP;</li> <li>3. Патерни проектування</li> <li>4. Розуміння принципів роботи веб-застосунку та протоколів TLS, SSL, HTTP(s);</li> <li>5. Розуміння LayerArchitecture;</li> <li>6. Знання ORM (EntityFramework).</li> <li>7. Архітектурні підходи до створення веб-застосунків</li> </ol>
<b>Що буде вивчатися</b>	<ol style="list-style-type: none"> <li>1. Архітектури розподілених та масштабованих систем</li> <li>2. Компоненти архітектури та критерії їх вибору</li> <li>3. Розгортання та розміщення інформаційних систем (CI/CD)</li> <li>4. Контейнеризація та оркестрація сервісів</li> <li>5. Способи оптимізації розподілених та масштабованих систем</li> </ol>
<b>Чому це цікаво/треба вивчати</b>	Аналіз пропозицій працевлаштування в IT-галузі на загальновідомих інтернет-агрегаторах дає змогу зрозуміти, що розробники програмного забезпечення з використанням платформи .NET займають одні з перших позицій щодо попиту на їх знання та вміння. Компетенції, що розглядаються в межах даного курсу є складовими частинами більшості вимог до кандидатів в межах вакансій на back-endaбо full-stackрозробника.
<b>Чому можна навчитися (результати навчання)</b>	<ol style="list-style-type: none"> <li>1. Проектувати архітектуру для масштабованих систем, обрати її компоненти, налаштувати та керувати етапами та процесом розміщення та розгортання;</li> <li>2. Виконувати задачі оптимізації веб-застосувань на різних архітектурних рівнях в межах системи.</li> </ol>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<p>Дисципліна надає набір вмінь та компетенцій, що є мінімально необхідними для спеціаліста з розробки програмного забезпечення, який бажає стати C#/.Net-розробником.</p> <p>Компетенції, які надає дисципліна:</p> <ul style="list-style-type: none"> <li>● Створення розподілених систем на мові C#, що працюють на платформах сімейства .NET;</li> <li>● Вміння застосовувати різні архітектурні підходи для оптимального вирішення поставлених задач.</li> </ul>
<b>Інформаційне забезпечення</b>	Силабус дисципліни, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
<b>Форма проведення занять</b>	Лекції та лабораторні (комп'ютерні практикуми)
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	ас.Вовк Є.А.

<b>Дисципліна</b>	<b>Системи автоматизованого проектування комп'ютерних систем</b>
<b>Кафедра</b>	Обчислювальної техніки
<b>Рівень ВО</b>	Перший (бакалаврський)

<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	4 курс, 7 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська, англійська
<b>Вимоги до початку вивчення</b>	Базові знання з вищої математики, програмування, дискретної математики, комп'ютерної схемотехніки, архітектури комп'ютерів.
<b>Що буде вивчатися</b>	<ul style="list-style-type: none"> <li>— основи будови систем автоматизованого проектування (САПР) комп'ютерних систем (КС);</li> <li>— технологія проектування спеціалізованих конвейерних КС на базі програмованих логічних інтегральних схем (ПЛІС) з використанням сучасних САПР;</li> <li>— проектування спеціалізованих багатопроцесорних КС та їх інтерфейсів.</li> </ul>
<b>Чому це цікаво/треба вивчати</b>	<ul style="list-style-type: none"> <li>— сучасні ПЛІС є чи не єдиною елементною базою, що дає змогу проектувати новітні КС, які відрізняються високими характеристиками продуктивності, надійності, енергоспоживання;</li> <li>— ПЛІС використовуються в медичинських приладах, телекомунікаціях, транспорті, датацентрах, інтернеті речей, військовій електроніці, аерокосмічній галузі, для вирішення задач штучного інтелекту;</li> <li>— є великий дефіцит професійних кадрів, як в Україні, так і в світі, які спроможні проектувати КС на ПЛІС, зарплатня яких, як правило, є вищою за зарплатню пересічного програміста.</li> </ul>
<b>Чому можна навчитися (результати навчання)</b>	<ul style="list-style-type: none"> <li>— складати паралельні алгоритми високопродуктивних обчислень,</li> <li>— програмувати алгоритми мовою VHDL для реалізації в апаратних КС,</li> <li>— моделювати і тестувати проекти КС, компілювати опис КС у прошивку ПЛІС.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>— проводити аналіз придатності ПЛІС для виконання комп'ютерних задач, які потребують високої продуктивності,</li> <li>— проектувати високопродуктивні КС на ПЛІС за ефективною технологією,</li> <li>— виконувати модернізацію існуючих високопродуктивних КС на ПЛІС,</li> <li>— організувати виконання алгоритмів, описаних мовами C, Python, на апаратних прискорювачах на ПЛІС.</li> </ul> <p>Володіння технологією програмування ПЛІС затребуване в Україні в галузях проектування військової техніки, телекомунікаціях, захисту інформації, авіабудування, в т.ч. проектування дронів і за кордоном у цих галузях, а також в системах штучного інтелекту, IoT, ADAS, робототехніці, проектуванні замовлених мікросхем.</p>
<b>Інформаційне забезпечення</b>	Силабус дисципліни, навчально-методичний комплекс.
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	д.т.н., професор Сергієнко А.М.

Дисципліна	Розробка програмного забезпечення. Проектний та програмний менеджмент.
Кафедра	Обчислювальної техніки
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 150
Курс	4, семестр 7.
Обсяг	4 кредити ЄКТС
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: Програмування, Інженерія програмного забезпечення.
Що буде вивчатися	В рамках курсу студенти зможуть отримати необхідні знання та навички в контексті організації процесу розробки програмних продуктів та фаз проектів (Solution design \ Discovery, Development, User Acceptance testing, Support etc.). Передбачається вивчення і практичне засвоєння наступних складових: вимоги до початку проекту, типи проектів та їх особливості; підходи до планування та звітності (Objectives and key results, etc.); позиції та відповідальність на різних ланках ІТ підрозділів. Курс також включає в себе вивчення кейсів успіху та випадків невдач у розробці програмного забезпечення для навчання на прикладах реального життя.
Чому це цікаво/треба вивчати	Розробка програмного забезпечення має динаміку швидкого розвитку та формування індустріальних стандартів не тільки з точки зору технологій, але й організації самого процесу. Тобто успішне виконання програмних проектів все більше залежить від ефективного керівництва, розуміння складових та використання кращих практик у організації розробки, що забезпечує досягнення високих результатів та задоволення потреб користувачів.
Чому можна навчитися (результати навчання)	Курс має за мету дати розуміння організації процесу розробки програмних продуктів, що є важливою складовою для всіх дотичних до цієї сфери.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Набуті знання можна використовувати під час розробки програмних продуктів будь якого типу та сфери застосування. Допоможе швидко адаптуватись в існуючому процесі або зайняти проактивну позицію в організації.
Інформаційне забезпечення	Силабус, РСО, конспект лекцій, завдання на лабораторні роботи
Форма проведення занять	Лекції, лабораторні роботи (практикуми)
Семестровий контроль	Залік
Викладач	Доц. Волокита Артем Миколайович, Project\Delivery manager DataArt Осмоловський Олександр

Дисципліна	Мережеве управління та протоколи
Кафедра	Інформатики та програмної інженерії

<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	4 курс, 7 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання курсів: «Основи програмування», «Теорія алгоритмів», «Основи комп'ютерних мереж», «інтернет-технології».
<b>Що буде вивчатися</b>	<ul style="list-style-type: none"> <li>• Види мереж.</li> <li>• Процес організації автоматизованих систем.</li> </ul> Мережеві протоколи.
<b>Чому це цікаво/треба вивчати</b>	Сучасність надає можливість швидкого та якісного використання інформаційно-технологічних мереж на базі протоколів.
<b>Чому можна навчитися (результати навчання)</b>	Якісне використання ресурсів в комп'ютерних мережах, універсальне застосування технологій на базі мережевих протоколів, навчитися роботі в команді.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>• володіння знаннями про типологію та особливості мережі</li> <li>• Побудова локальних та глобальних комп'ютерних мереж;</li> <li>• Застосування протоколів в комп'ютерних мережах</li> </ul> Застосування базових технологій розробки мережевих протоколів на базі стандартів еталонних моделей OSI/ISO
<b>Інформаційне забезпечення</b>	Силабус дисципліни, онлайн документація та альтернативні онлайн лекції
<b>Форма проведення занять</b>	Лекції, лабораторні роботи (комп'ютерний практикум)
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	доц., к.т.н. Зенів І.О.

## 4 курс 8 семестр

### Освітній компонент 12Ф-каталогу

Дисципліна	Алгоритми та математичні методи захисту інформації
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	4 курс, 8 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін «Програмування», «Вища математика», «Захист інформації».
Що буде вивчатися	Алгоритми та математичні методи захисту інформації.
Чому це цікаво/треба вивчати	Криптографія - це наука, яка займається будівництвом безпечних шифрів, тобто алгоритмів, що забезпечують перетворення електронних документів в зовні випадковий набір символів, з яких можна відновити вихідний документ, тільки знаючи якийсь пароль (секретне слово). Криптографія розвивалася з самого початку історії людства. Відомі приклади шифрування текстів з глибокої давнини - у римлян, греків та інших народів. Проте ера сучасної криптографії почалася порівняно нещодавно, в 70-і роки ХХ століття. В межах даної навчальної дисципліни розглядається навчальний матеріал, що має відношення до основ побудови сучасних алгоритмів криптографії та пов'язаний з ними.
Чому можна навчитися (результати навчання)	Студент повинен після засвоєння матеріалу навчальної дисципліни: <b>знати:</b> - концепції інформаційної безпеки, принципів безпечного проектування ІС а ІТ, методології безпечного програмування, погроз і атак, безпеки комп'ютерних мереж, методи криптографії. <b>вміти:</b> - зберігати конфіденційність, цілісність та доступність інформації, забезпечувати автентичність, відстежуваність та надійність інформації в умовах неповноти та невизначеності вихідних даних, багатокритеріальності професійних задач.
Як можна користуватися набутими знаннями і вміннями (компетентності)	Застосовувати методи та засоби забезпечення інформаційної безпеки, розробляти та експлуатувати спеціальне програмне забезпечення захисту інформаційних ресурсів об'єктів критичної інформаційної інфраструктури.
Інформаційне забезпечення	Навчальна і робоча програми дисципліни (з РСО), опорний конспект лекцій, методичні вказівки до виконання лабораторних робіт, перелік завдань для самостійної роботи студента, перелік питань до МКР та семестрового контролю.
Форма проведення занять	Лекції, лабораторні роботи.
Семестровий контроль	Залік
Викладач	к.т.н., доцент Сокульський О.Є.

<b>Дисципліна</b>	<b>Управління виробництвом на базі застосування систем класу ERP</b>
Кафедра	Інформатики та програмної інженерії
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	4 курс, 8 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Відомості про застосування інформаційних систем в бізнесі Відомості про виробничі бізнес системи
Що буде вивчатися	* Теоретичні та практичні аспекти побудови виробничих бізнес-систем * Стандарти управління виробництвом: MRP-II, LeanProduction, TOC Практика застосування інформаційних систем для автоматизації процесів у виробництві.
Чому це цікаво/треба вивчати	Виробничі системи як напрямок автоматизації завжди будуть знаходитись на передньому краї застосування ІТ-рішень в бізнесі. Сучасні стандарти управління виробництвом є основою побудови ІС у вигляді ERP-додатків
Чому можна навчитися (результати навчання)	Матеріали курсу поєднують бізнес дослідження в сфері управління виробничими системами з аналітичними ІТ-дослідженнями побудови ERP-систем. Зрозуміння цих моментів створює фундаменальний базис для використання ІТ-технологій програмування та інформаційного проектування на практиці.
Як можна користуватися набутими знаннями і уміннями (компетентності)	Розробники програмного та інформаційного забезпечень можуть виконувати функції та роботи аналітичного характеру, приєднуватись до практичної роботи на більш ранніх стадіях виконання ІТ проектів.
Інформаційне забезпечення	Навчальна та робоча програми дисципліни, РСО, онлайн документація
Форма проведення занять	Лекції та лабораторні
Семестровий контроль	Залік
Викладач	доц., к.т.н. Новінський В.П.

<b>Дисципліна</b>	<b>Оптимізація та балансування навантажень у базах даних</b>
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 150
Курс, семестр	4 курс, 8 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін: • досвід у використанні БД; вміння встановлювати,



	<p>підключатися до БД, розуміння синтаксису SQL та бажано досвід у використанні PostgreSQL</p> <ul style="list-style-type: none"> <li>• написання запитів до БД з використанням декількох таблиць</li> <li>• мінімальні знання по архітектурі ПК та комп'ютерним мережам</li> <li>• базові знання по віртуалізації та, зокрема, робота з Docker</li> </ul> <p>знання алгоритмів пошуку та роботи з деревами</p>
<b>Що буде вивчатися</b>	<ul style="list-style-type: none"> <li>• вивчення різних типів БД, включно з аналізом реляційних БД та PostgreSQL БД</li> <li>• основні аспекти, використання БД, розбір проблеми нецільового використання окремих типів БД у IT-інфраструктурі</li> <li>• роль індексації у БД, створення власних індексів</li> <li>• CAP-теорема та її роль при проектуванні власних продуктів з урахуванням використання БД</li> <li>• оптимізація запитів до БД – використання процесорного часу, пам'яті та кешу</li> <li>• бекапи та снєпшоти – задачі, що вони вирішують та аспекти їх використання</li> <li>• міграція БД</li> </ul>
<b>Чому це цікаво/треба вивчати</b>	<p>На сьогодні більшість програмних застосунків використовують бази даних. Здебільшого розробники використовують ту БД, до якої звикли. За такого підходу не враховуються специфіка задач, що вирішують обрані БД у програмних продуктах. Мета курсу:</p> <ul style="list-style-type: none"> <li>• продемонструвати особливості кожного типу БД (їх переваги та недоліки)</li> <li>• продемонструвати доречність використання окремих типів БД</li> <li>• навчити оптимізації запитів у реляційні БД для зменшення навантаження на сервери</li> <li>• навчити раціонально використовувати кеш</li> <li>• підвищення надійності систем, до використовують БД шляхом бекапу даних та горизонтального розширення сервісів</li> </ul>
<b>Чому можна навчитися (результати навчання)</b>	<ul style="list-style-type: none"> <li>• оптимізація запитів до БД</li> <li>• проектування власної IT-інфраструктури для вирішення задач при високому навантаженні</li> <li>• підвищення надійності IT-інфраструктур при використанні БД</li> <li>• балансування навантаження у IT-інфраструктурах на прикладі сервісу БД</li> <li>• основи міграції БД</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<p>Після проходження курсу, студент буде:</p> <ul style="list-style-type: none"> <li>• розуміти різницю між реляційними та нереляційними БД</li> <li>• вміти проектувати архітектуру ПП (програмний продукт) з урахуванням задач, що цей ПП виконує <ul style="list-style-type: none"> <li>• доцільно використовувати процесорний час, кеш декількох рівнів</li> <li>• розуміти переваги HDD над що SSD</li> </ul> </li> </ul>
<b>Інформаційне забезпечення</b>	Навчальна і робоча програми дисципліни, PCO, навчально-методичний комплекс
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., доцент Галушко Д.О.

<b>Дисципліна</b>	<b>Програмування смарт-контрактів</b>
<b>Кафедра</b>	Інформаційних систем та технологій
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	4 курс, 8 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дискретної математики, інформатики, архітектури комп'ютерних систем та мереж, теорії алгоритмів та автоматів, мов та технологій програмування, технології блокчейн
<b>Що буде вивчатися</b>	Підходи, засоби та технології програмування смарт-контрактів. Життєвий цикл смарт-контрактів. Програмування смарт-контрактів мовою Solidity на блокчейні Ethereum. Мова програмування Rust для проєктування смарт-контрактів на блокчейні Near. Операційна система блокчейнів BOS. Конструювання смарт-контрактів. Приклад смарт-контракту голосування. Юридична легітимність смарт-контрактів.
<b>Чому це цікаво/треба вивчати</b>	Основою сучасних детермінованих інформаційних систем є web3 децентралізовані додатки, рівнем виконання яких є смарт-контракти на блокчейнах. Оволодіння навичками та знаннями програмуванням смарт-контрактів дозволить проєктувати web3-орієнтовані децентралізовані додатки для освітніх, виробничих та бізнес задач.
<b>Чому можна навчитися (результати навчання)</b>	Принципи розробки смарт-контрактів для сучасних блокчейнів, зокрема, на прикладах блокчейнів Ethereum та Near Protocol. Мови програмування блокчейнів й смарт-контрактів Solidity та Rust. Програмування прикладу смарт-контракту та інтерфейсу децентралізованого додатку системи демократичного голосування. Виклик методів смарт-контракту через CLI. Моніторинг роботи смарт-контрактів засобами мережевих провідників.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Програмування смарт-контрактів на Solidity для EVM-блокчейнів. Програмування смарт-контрактів на Rust для блокчейну Near Protocol. Програмна реалізація користувацьких алгоритмів децентралізованих додатків. Тестування та розгортання смарт-контрактів в блокчейні. Розробка інтерфейсу для децентралізованих додатків.
<b>Інформаційне забезпечення</b>	Силабус, Презентації лекцій, контрольні тести
<b>Форма проведення занять</b>	Лекції, лабораторний (комп'ютерний) практикум
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	ст. викладач Яланецький Валерій Анатолійович

Дисципліна	Комп'ютерна лінгвістика
Кафедра	Інформатики та програмної інженерії
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	4 курс, 8 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	- Базові знання програмування - Базові знання теорії ймовірності та мат. статистики
Що буде вивчатися	- Види аналізу тексту - Математичне та інформаційне забезпечення лінгвістичного аналізу - Методи визначення параметрів тексту та особливих рис авторів
Чому це цікаво/треба вивчати	На теперешній час та надалі комп'ютерна лінгвістика буде відігравати все більш значну роль при аналізі інформаційних потоків. Використання на базі методів комп'ютерної лінгвістики методів машинного навчання, забезпечення вхідної інформації для роботи нейронних мереж, різноманітність задач, що потребують вирішення - все це робить дану дисципліну однією з найзатребуваних для сучасного ІТ-спеціаліста, що спрямовує свою діяльність на роботі з інформацією.  Коло задач комп'ютерної лінгвістики коливається від пошуку подібних документів (задачі пошукових систем, плагіату тощо), визначення тональності текстів для аналізу вподобань (цільова реклама, системи відгуків, соціальні науки) до виявлення прихованих рис, що дозволяють визначити параметри автору тексту, вилучення фактажу з тексту для аналізу, побудови психологічного портрету автора, аналізу соціальних мереж та медіа.
Чому можна навчитися (результати навчання)	1) виділення рівнів аналізу тексту, типовим методам вирішення практичних задач, особливостям використання різних методів аналізу; 2) методам вирішення типових задач, розбиття (декомпозиції) складних задач аналізу тексту.
Як можна користуватися набутими знаннями і уміннями (компетентності)	- використовувати методи та підходи комп'ютерної лінгвістики для вирішення практичних задач; - використовувати та досліджувати методи збору інформації, фільтрації інформації. - аналізувати інформаційні потоки на предмет емоційного забарвлення
Інформаційне забезпечення	навчальна та робоча програма дисципліни, РСО, навчальний посібники та підручники (електронні видання)
Форма проведення занять	Лекції, лабораторні заняття
Семестровий контроль	Залік
Викладач	к.т.н, доцент Фіногенов О.Д.

Дисципліна	Системи безпеки програм і даних
Кафедра	Обчислювальної техніки
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	<b>Окрім тих студентів що вивчали дану дисципліну на 3-му курсі!!!</b> Кількість студентів на курс від 25 до 120
Курс, семестр	4 курс, 8 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з архітектури комп'ютера, комп'ютерних мереж, алгоритмів та структур даних, основ програмування, дискретної математики, теорії ймовірності та математичної статистики.
Що буде вивчатися	Цей курс присвячений поглибленому вивченню окремих питань щодо математичних та алгоритмічних аспектів захисту інформації у комп'ютерних системах та проектуванню та розробці захищеного програмного забезпечення та систем захисту. У такому курсі вивчаються методи і алгоритми захисту при передачі та зберіганні даних, а також вивчаються питання розробки програмно-апаратних засобів захисту. Зокрема, захищені мережні протоколи, методи шифрування, криптографії, стеганографії. Питання створення та організації захищених сервісів з використанням хмарних технологій у різноманітних галузях, зокрема, електронної комерції, електронного банкінгу, функціонування криптовалют.
Чому це цікаво/треба вивчати	Сучасні реалії функціонування цифрового суспільства вимагають використання та розробки комп'ютерних засобів та систем, стійких до несанкціонованого доступу та використання програм та даних. Кожний кваліфікований фахівець, який має відношення до програмного забезпечення повинен знати принципи інформаційної безпеки та методи захисту програм і даних та вміти їх грамотно застосовувати.
Чому можна навчитися (результати навчання)	Навчитися приймати рішення щодо вибори структури засобів захисту даних; розумітися на математичних методах і алгоритмах криптографії і шифрування даних. Розробляти програмні засоби захисту інформації, стійкі до хакерських атак. Організувати захищене зберігання, обробку та передавання інформації з використанням технологій розподілених та хмарних систем.
Як можна користуватися набутими знаннями і вміннями (компетентності)	Здатність аналізувати, вибирати, розробляти і застосовувати методи і засоби для забезпечення інформаційної безпеки, кібербезпеки. Можливість виконувати роботи у якості аналітика комп'ютерних систем та програмного забезпечення. Бути кваліфікованим розробником програмного забезпечення та користувачем програмно-апаратних комплексів.
Інформаційне забезпечення	Силабус дисципліни, навчально-методичний комплекс.
Форма проведення занять	Лекції, лабораторні роботи, самостійна робота
Семестровий контроль	Залік
Викладач	к.т.н., доцент Волокита А.М.

Дисципліна	Програмування комп'ютерної графіки
Кафедра	Обчислювальної техніки
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс	3-й, 6 семестр
Обсяг	4 кредити ECTS
Мова викладання	Українська, англійська
Кафедра	Обчислювальної техніки
Вимоги до початку вивчення	Базові знання з елементарної математики, аналітичної геометрії. Розуміння структур даних та алгоритмів. Вміння створювати програмні застосунки мовами C/C++ або C# або Java (можливий індивідуальний вибір мови для кожного студента).
Що буде вивчатися	<p>Базові поняття комп'ютерної графіки</p> <ul style="list-style-type: none"> <li>- Растровий та векторний підходи. Особливості використання кожного з них при програмуванні власних програмних засобів</li> <li>- Колірні моделі та кодування кольорів</li> <li>- Базові алгоритми rasterізації ліній, кривих, фігур з заповненням – полігонів та контурів довільної форми</li> <li>- Системи координат, проекції 3d-сцен, перетворення систем координат, трансформування об'єктів</li> <li>- Анімація</li> </ul> <p>Просунуті аспекти:</p> <ul style="list-style-type: none"> <li>- Відомі моделі опису просторових об'єктів і сцен: полігональні, воксельні, фрактальні та інші. Перетворення моделей</li> <li>- Особливості організації структур даних для підтримки у власних програмах рендерингу тривимірних сцен відповідно вимог потрібного рівня деталізації та (або) якості відображення. Оптимізація по швидкодії рендерингу та (або) заощадження ресурсів</li> <li>- Особливості відображення об'єктів відносно поверхні Землі. Системи координат, проекції. Основні аспекти реалізації відображення у геоінформаційних системах (GIS).</li> <li>- Геометричне моделювання кінематики складних рухомих об'єктів. Особливості створення зображення рухомих фігур людей, тварин і реалізація цього у власних програмах</li> </ul> <p>Інтерфейси програмування відеокарт та графічних процесорів:</p> <ul style="list-style-type: none"> <li>- OpenGL, OpenGL ES</li> <li>- DirectX Graphics</li> </ul> <p>Програмування власних програмних застосунків комп'ютерної 3d-графіки:</p> <ul style="list-style-type: none"> <li>- для десктопних комп'ютерів - з використанням мови C/C++ або C# або Java (індивідуальний вибір кожного студента).</li> <li>- для пристроїв на платформі Android - з використанням Java або Kotlin (індивідуальний вибір кожного студента).</li> </ul>
Чому це цікаво/треба вивчати	Знання, вміння та навички створення програмних рішень з комп'ютерної графіки дозволяють успішно виконувати ІТ-проекти різноманітних предметних галузей. Без знань в галузі комп'ютерної графіки не зможе обійтися будь-який сучасний фахівець комп'ютерних технологій.
Чому можна	Вивчити основи, математичні методи і алгоритми сучасної

<b>навчитися (результати навчання)</b>	<p>комп'ютерної графіки.</p> <p>Ознайомитися з методами побудови реалістичних зображень тривимірних сцен, моделюванням матеріальних властивостей об'єктів, урахуванням законів оптики тощо.</p> <p>Ознайомитися з основними положеннями та навчитися програмувати засоби геометричного моделювання об'єктів тривимірних сцен а також для моделювання кінематики складних рухомих об'єктів.</p> <p>Ознайомитися з особливостями реалізації відображення об'єктів в геоінформаційних системах.</p> <p>Навчитися створювати власний програмний рушій для високошвидкісного рендерингу 3d сцен з використанням АРІ графічних прискорювачів як для десктопних систем, так і для гаджетів Android.</p> <p>Навчитися створювати власний програмний рушій для високоякісного Raytracing-рендерингу 3d сцен.</p>
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	<p>Набуті знання, уміння та практичні навички дозволять успішно реалізовувати складні проекти у будь-якій галузі – усюди, де необхідна комп'ютерна графіка.</p> <p>Програмні активи успішно виконаних завдань лабораторних робіт студенти зможуть надалі розвивати, вдосконалювати задля використання у власних майбутніх проектах.</p> <p>Фахівець, який здатен самостійно запрограмувати якісне відображення складних об'єктів та сцен, зможе не тільки розібратися з вже існуючими зразками певних засобів подібного призначення – він може створювати власні технології. А такі фахівці, як правило, цінуються найвище</p>
<b>Інформаційне забезпечення</b>	<p>Силабус, навчально-методичний комплекс</p>
<b>Форма проведення занять</b>	<p>Лекції, лабораторні роботи, самостійна робота</p>
<b>Семестровий контроль</b>	<p>Залік</p>
<b>Викладач</b>	<p>Доцент, к.т.н. Порєв В.М.</p>

## Освітній компонент 13 Ф-каталогу

Дисципліна	Технології штучного інтелекту
<b>Кафедра</b>	Інформаційних систем та технологій
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 150
<b>Курс, семестр</b>	4 курс, 8 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: математичний аналіз, дискретна математика, об'єктно – орієнтоване моделювання, теорія алгоритмів, теорія автоматичного управління, моделювання систем, веб-технології та веб дизайн, програмування.
<b>Що буде вивчатися</b>	Поняття штучного інтелекту. Поняття інтелектуальної системи (ИС) та інтелектуальної задачі (ІЗ). Способи подання інтелектуальної задачі та методи пошуку рішень. Пошук рішень ІЗ у просторі станів. Знання та моделі представлення знань, продукційні моделі представлення знань у США. Управління пошуком рішень у продукційних системах. Класичні технології ШІ (машинне навчання, глибоке навчання, комп'ютерний зір, обробка природньої мови). Експертні системи (ЕС): призначення та принципи побудови; узагальнена архітектура; класи задач, які вирішуються за допомогою ЕС. Розробка ЕС: етапи розробки; придбання знань; пошук та пояснення рішень. Інженерія знань. Онтологічний підхід до представлення та інтеграції знань у розподілених інформаційних середовищах типу Інтернет.
<b>Чому це цікаво/треба вивчати</b>	Технології штучного інтелекту використовуються практично у всіх сферах, включаючи ІТ. Фахівці здатні вирішувати складні задачі, пов'язані зі створенням та навчанням штучних нейронних мереж, розпізнанням мови та зображень, аналізу великого обсягу даних, управляти складними системами і системами різного призначення.
<b>Чому можна навчитися (результати навчання)</b>	Результатами вивчення дисципліни є знання: основних підходів до вирішення інтелектуальних задач, основних понять та означень штучного інтелекту, технологій, підходів і методів штучного інтелекту, способів подання інтелектуальної задачі та методів пошуку рішень, моделей представлення знань у системах із ШІ, сучасних тенденцій та підходів до створення систем із ШІ.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Студенти можуть досліджувати й оцінювати програмні продукти з алгоритмами штучного інтелекту, розробляти нейромережі різної архітектури, зокрема: перцептрон, неокогнітрон, когнітрон, мережі Хопфілда та Хемінга, розробляти системи розпізнавання образів, застосовувати класичні технології штучного інтелекту для вирішення інтелектуальних задач, модулювати структуру та розробляти експертні системи.
<b>Інформаційне забезпечення</b>	Конспект лекцій - <a href="https://campus.kpi.ua/student/index.php?mode=mob&amp;show&amp;irid=202438">https://campus.kpi.ua/student/index.php?mode=mob&amp;show&amp;irid=202438</a> Презентації - <a href="https://campus.kpi.ua/student/index.php?mode=mob&amp;show&amp;irid=202439">https://campus.kpi.ua/student/index.php?mode=mob&amp;show&amp;irid=202439</a>

	силабус курсу, методичні вказівки по виконанню лабораторного практикуму, контрольні завдання.
<b>Форма проведення занять</b>	Лекції та комп'ютерні практикуми
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	ст. викладач Польшакова О.М.

<b>Дисципліна</b>	<b>Інформаційна складова систем радіокерування</b>
<b>Кафедра</b>	Інформаційних систем та технологій
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 150
<b>Курс, семестр</b>	4 курс, 8 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін «Програмування», «Вища математика».
<b>Що буде вивчатися</b>	Принципи побудови систем радіокерування, промислових систем, наземних рухомих платформ, повітряних радіокерованих моделей. Типові структурні і функціональні схеми систем радіо керування для промислових систем, наземних рухомих платформ, повітряних радіокерованих моделей на основі програмованих модулів неліцензованих частот (2,5 ГГц, 433 МГц); Програмні і апаратні складові каналів систем радіокерування для на основі модулів NRF24 (аналогів). Програмне управління на базі STM32F103 (STM32F411). Технології створення, тестування і супроводження програмних систем радіокерування на основі мови програмування C(C++). Програмні і апаратні складові контролерів систем радіокерування на основі мінікомп'ютерів OrangePi (RaspberryPi). Типові структурні і функціональні схеми систем радіокерування для FPV дронів, радіокерованих моделей літаків.
<b>Чому це цікаво/треба вивчати</b>	Тому, що системи програмованого радіокерування на базі модулів NRF24 та програмної складової на основі STM32F103 (STM32F411) актуальні на теперішній час. Програмні і апаратні складові контролерів систем радіокерування для на основі мінікомп'ютерів OrangePi (RaspberryPi) отримали широкий розвиток і продовжують вдосконалюватися.
<b>Чому можна навчитися (результати навчання)</b>	Будувати типові структурні і функціональні схеми систем радіокерування для промислових систем, наземних рухомих платформ, повітряних радіокерованих моделей. Обирати елементну базу і доступні функціональні складові програмованих радіомодулів на частотах (2,5 ГГц, 433 МГц); Розробляти, тестувати і супроводжувати програми (мова програмування C(C++)) для інформаційної складової систем радіокерування на основі STM32F103 (STM32F411). Розробляти, тестувати і супроводжувати програми (мова програмування C(C++) або за вибором студента)



	для інформаційної складової систем радіокерування на основі OrangePi (Raspberry Pi). Самостійно розробляти структурні і функціональні схеми, здійснювати крупноблокову збірку радіокерованих FPV дрони з використанням доступних елементів та інших радіокерованих систем.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Застосування в професійній діяльності в якості розробників програмних складових інформаційних систем, науковій діяльності з досліджень інформаційної складової радіокерування.
<b>Інформаційне забезпечення</b>	Навчальна і робоча програми дисципліни (з РСО), опорний конспект лекцій, методичні вказівки до виконання лабораторних робіт, перелік завдань для самостійної роботи студента, перелік питань до МКР та семестрового контролю.
<b>Форма проведення занять</b>	Лекції, лабораторні роботи.
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., доцент Катін П.Ю.

<b>Дисципліна</b>	<b>Програмування вбудованих систем на основі архітектури ARM</b>
<b>Кафедра</b>	Інформаційних систем та технологій
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	4 курс, 8 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання: основ програмування (C, C++ або C#) і алгоритмізації. Бажані знання основ мікроелектроніки, дискретної математики.
<b>Що буде вивчатися</b>	Способи аналізу функціональних вимог, структури і технології вбудованих систем, приклади використання у автомобільній індустрії; - структурна, функціональна схема вбудованої системи на базі мікроконтролера (у першому наближенні ARM, Cortex-M (STM32F10x)); - види сучасних програмних технологій програмування C (C++); - досконале вивчення обраної технології програмування, для студентів з досвідом дозволено самостійно обирати технологію; - операційні системи реального часу для вбудованих систем; - вбудована система як елемент автомобільного автопілоту; - основи бізнесу у індустрії вбудованих систем.
<b>Чому це цікаво/треба вивчати</b>	За один семестр можна вивчити базові основи програмування і розробки вбудованих систем. Отримати перший досвід, визначитись з потенційним напрямком роботи у цій сфері: - бізнес-аналітик у сфері розробки вбудованих систем, у тому числі у автомобільній сфері; - інженер-розробник програмної і апаратної складових вбудованих систем, у тому числі у автомобільній сфері;

	- тестер (проектувальник вбудованих систем), та інші споріднені професійні напрямки. Визначитись з питаннями початку власного бізнесу.
<b>Чому можна навчитися (результати навчання)</b>	Основний напрям - програмування вбудованої системи, розробка програмної частини При цьому надаються вміння, досвід і навички проектування і бізнес-аналітики вбудованих систем (на прикладі автомобільної сфери). Повному циклу розробки вбудованих систем. Самостійному і набуттю знань з питань проектування і програмування вбудованої системи.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Отриманні знання, вміння, досвід і навички дають можливість працевлаштування у відповідній професійній сфері. Також отриманий досвід дозволяє зробити підґрунтя щодо створення власного бізнесу у галузі вбудованих систем.
<b>Інформаційне забезпечення</b>	Силабус, методичні рекомендації, підтримка у Телеграмі (група відкривається на час занять). Лабораторія ауд. 517-18, що має все для розробки (у першому наближенні ARM, Cortex-M (приклад STM32F10x)), середовище розробки <a href="https://www.keil.com">https://www.keil.com</a> .
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., доц. Ковальов М.О.

Дисципліна	Інженерія знань
<b>Кафедра</b>	Інформатики та програмної інженерії
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	4 курс, 8 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Дисципліна «Інженерія знань» базується на наступних дисциплінах – «Дискретна математика», «Теорія ймовірностей», «Функціональне програмування», «Мультипарадигмене програмування». З практичного боку бажані знання мов штучного інтелекту LISP, PROLOG, системи CLIPS.
<b>Що буде вивчатися</b>	– загальні методи представлення знань у системах штучного інтелекту, починаючи з попереднього досвіду класичної логіки та доведення теорем, потім звертається до нових підходів, які поширюють класичну логіку - наприклад, для обробки якісної або невизначеної інформації - та для покращення її обчислювальної придатності; – розглядаються особливі проблеми репрезентації та аргументування деяких основних сфер знань, включаючи час, простір, причинно-наслідкові зв'язки та дії; – досліджуються важливі програми подання знань та міркувань, області застосування яких охоплюють широкий спектр штучного

	інтелекту, включаючи відповіді на запитання, семантичні мережі, планування, робототехніку та мультиагентні системи.
<b>Чому це цікаво/треба вивчати</b>	<ul style="list-style-type: none"> <li>- представлення знань та міркування лежить в основі великого виклику штучного інтелекту;</li> <li>- зрозуміти природу інтелекту та пізнання настільки добре, що комп'ютери можуть бути створені для виявлення людських здібностей.</li> </ul>
<b>Чому можна навчитися (результати навчання)</b>	<ul style="list-style-type: none"> <li>- застосовувати існуючі методи та алгоритми представлення знань для розв'язання практичних задач штучного інтелекту</li> <li>застосовувати методи розробки алгоритмів, конструювання експертних систем та структур знань</li> </ul>
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>- застосовувати моделі інженерії знань для розробки програмного забезпечення систем, що базуються на знаннях (knowledge base systems)аналізувати, цілеспрямовано шукати і вибирати необхідні засоби інженерії знань для опису, конструювання, впровадження баз знань предметних областей</li> </ul>
<b>Інформаційне забезпечення</b>	Силабус дисципліни, лекційні матеріали, лабораторні завдання
<b>Форма проведення занять</b>	Лекції та лабораторні (комп'ютерні практикуми, розрахунково-графічна робота)
<b>Семестровий контроль</b>	Залік
<b>Додаткові дані</b>	<p>25 тем дисципліни, організовані у три розділи «Загальні методи», «Спеціалізовані представництва» та «Застосування», забезпечують унікальний огляд найкращих результатів, котрі досягла інженерія знань. Ми сподіваємось, що студенти, дослідники та практики у всіх сферах штучного інтелекту та когнітивної науки знайдуть цей курс корисним ресурсом.</p> <p>Частина тем дисципліни може бути винесена на самостійне вивчення.</p>
<b>Викладач</b>	доц., к.т.н. Баклан І.В.

<b>Дисципліна</b>	<b>Сучасні методи визначення вимог до ІТ-рішень</b>
<b>Кафедра</b>	Інформатики та програмної інженерії
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 90
<b>Курс, семестр</b>	4 курс, 8 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисципліни «Компоненти програмної інженерії. Частина 2. Моделювання та аналіз вимог до програмного забезпечення»
<b>Що буде вивчатися</b>	Сучасні підходи до визначення функціональних та нефункціональних вимог, підходи до побудови архітектури вимог до програмного забезпечення, вплив контексту проекту на вибір підходів та технік визначення вимог.
<b>Чому це цікаво/треба вивчати</b>	Провідну роль в проектах по розробці інформаційних технологій відіграють інженерія вимог та бізнес-аналіз, що використовується для виявлення потреб бізнесу, визначення найкращого варіанту рішення,

	та організації комунікації між зацікавленими сторонами проєкту. Враховуючи, що вимоги до програмного забезпечення є ключовим результатом роботи бізнес-аналітика і базисом для подальшої розробки та тестування ІТ-рішень, вибір оптимальних методів для визначення вимог стає ключовим фактором успіху ІТ проєктів.
<b>Чому можна навчитися (результати навчання)</b>	Дисципліна надає набір вмінь та компетенцій, що є мінімально необхідними для студента, який бажає в майбутньому стати інженером з вимог або бізнес-аналітиком: <ul style="list-style-type: none"> <li>• Знати сучасні методи визначення вимог до ІТ-рішень для предиктивного, адаптивного та гібридного підходів;</li> <li>• Знати методи побудови архітектури вимог для ІТ-рішень;</li> <li>• Визначати і документувати вимоги, що відповідають загальноприйнятим критеріям якості</li> <li>• Обирати техніки роботи з вимогами з врахуванням контексту проєкту.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>• Планувати підхід до бізнес-аналізу для ІТ-проєктів</li> <li>• Планувати підхід до керування вимогами</li> <li>• Проводити аналіз вимог та створювати специфікації вимог відповідно до життєвого циклу програмного забезпечення.</li> </ul>
<b>Інформаційне забезпечення</b>	Силабус дисципліни, лабораторний практикум, презентації лекційного матеріалу, контрольні завдання
<b>Форма проведення занять</b>	Лекції, лабораторні роботи.
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., доцент Гобов Д.А.

Дисципліна	<b>Мережеве програмування в середовищі Unix</b>
<b>Кафедра</b>	Обчислювальної техніки
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	4 курс, 8 семестр
<b>Обсяг</b>	4 кредити ЄКТС
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Вміння програмувати на С або С++ та вміння працювати в Unix-подібній операційній системі на рівні користувача. Базові знання з дисциплін Програмування, Структури даних та алгоритми, Комп'ютерні мережі.
<b>Що буде вивчатися</b>	Вивчатиметься мережеве програмування в середовищі Unix, тобто розроблення програм, які працюють із мережею, для Unix-подібних операційних систем на рівні використання системних викликів для взаємодії з ядром операційної системи. Надається ґрунтовна інформація про функції POSIX (та достатня інформація про реалізації відповідних системних викликів), які використовуються в програмах, які працюють із мережею. Дисципліна не орієнтована на мережеве програмування в якійсь конкретній реалізації Unix-подібної операційної системи, вивчатиметься переносне мережеве програмування. Дисципліна складається з таких тем: мережеві адреси

	та імена, TCP сокети, UDP сокети, SCTP сокети, комунікаційний домен Unix, мультиплексування введення-виведення, асинхронні дії, реалізації серверних програм різними способами.
<b>Чому це цікаво/треба вивчати</b>	Цю дисципліну доцільно вивчати тим, хто розроблятиме програми, які працюють із мережею, для Unix-подібних операційних систем на рівні використання системних викликів. Програмування завдань виконується мовою програмування C або C++, але набуті знання будуть корисні для вирішення деяких задач, які пов'язані з мережевим програмуванням, для Unix-подібних операційних систем іншими мовами програмування.
<b>Чому можна навчитися (результати навчання)</b>	Розробляти програми, які працюють із мережею, для Unix-подібних операційних систем на рівні використання системних викликів мовою програмування C або C++, створювати серверні програми різними способами.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Набуті знання можна використовувати під час розроблення програм, які працюють із мережею, для Unix-подібних операційних систем, для супроводження вихідного коду наявних програм, які працюють із мережею, для Unix-подібних операційних систем.
<b>Інформаційне забезпечення</b>	Силабус, РСО, конспект лекцій, завдання на лабораторні роботи
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	Ст. викладач Сімоненко Андрій Валерійович

<b>Дисципліна</b>	<b>Гібридні комп'ютерні системи</b>
<b>Кафедра</b>	Обчислювальної техніки
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	4 курс, 8 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: Теорія ймовірності та математична статистика, Системне програмування, Системне програмне забезпечення, Комп'ютерні системи, Архітектура комп'ютерів, Комп'ютерні мережі
<b>Що буде вивчатися</b>	Основи проектування аналогових та гібридних операційних блоків, аналогово-цифрових та цифро-аналогових перетворювачів інформації, аналогових та гібридних процесорів, методи математичного моделювання з допомогою операційних блоків, методи підготовки задач для аналогових та гібридних систем, методи програмування для аналогових та гібридних систем для спеціалістів фахового напрямку "Комп'ютерна інженерія", забезпечити достатню глибину засвоєння дисциплін основної інженерної підготовки зі спеціальності, а також для оволодіння сучасними технічними засобами та способами програмування аналогових та гібридних систем для наступних спеціалізації.

<b>Чому це цікаво/треба вивчати</b>	Отримані знання дозволять оволодіти основними методами побудови нетрадиційних комп'ютерних систем
<b>Чому можна навчитися (результати навчання)</b>	Проектуванню систем з гетерогенною структурою, наприклад систем з <a href="#">CPU</a> і <a href="#">GPU</a>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Після проходження курсу студенти зможуть проектувати високопродуктивні комп'ютерні системи
<b>Інформаційне забезпечення</b>	Силабус дисципліни, навчально-методичний комплекс.
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., доцент Селіванов В.Л.

<b>Дисципліна</b>	<b>Проектування вбудованих систем</b>
<b>Кафедра</b>	Обчислювальної техніки
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	4 курс, 8 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Необхідні дисципліни: «Вступ до операційної системи Linux», «Комп'ютерна логіка», «Архітектура комп'ютерів. Частина 1. Управляючі та арифметичні пристрої», «Архітектура комп'ютерів. Частина 2. Процесори», «Програмування», «Алгоритми і структури даних»
<b>Що буде вивчатися</b>	Методи перетворення проектної документації в реальний код. Способи якими оптимізації процесу написання коду для того щоб виконати поставлені задачі вчасно trade-off стратегії, робота з ризиками проекту і тд. Навички розподілення задач в часі та планування повного циклу розробки. Тестування коду Розгортання коду на серверній інфраструктурі.
<b>Чому це цікаво/треба вивчати</b>	Дисципліна "Проектування вбудованих систем" призначена для ознайомлення студентів з основами вбудованих систем та їх застосування в різних галузях. Вивчення цієї дисципліни дозволить студентам зрозуміти концепцію вбудованих систем, їх роль у сучасному технологічному середовищі та основні принципи їх роботи та розробки включаючи особливості проектування програмного та апаратного забезпечення. Вивчення цієї дисципліни допоможе студентам отримати необхідні базові знання та уміння для подальшого вивчення та професійного застосування в області вбудованих систем.
<b>Чому можна навчитися</b>	За результатами вивчення навчальної дисципліни "Проектування

<b>(результати навчання)</b>	<p>вбудованих систем" мають бути отримані такі знання.</p> <ul style="list-style-type: none"> <li>– Знання про основні принципи та концепції вбудованих систем, включаючи їх архітектуру, специфіку функціонування та обмеження.</li> <li>– Знання про особливості проектування ефективного та безпечного програмного та апаратного забезпечення.</li> <li>– Знання про різновиди мікроконтролерів, їх особливості, підходи до проектування програмного забезпечення.</li> <li>– Знання про методи відлагодження та порядок виконання програми з урахуванням особливостей вбудованих систем.</li> <li>– Знання про різні інтерфейси, протоколи передачі даних, їх особливості застосування.</li> </ul> <p>Уміння, які мають бути отримані у рамках вивчення навчальної дисципліни "Проектування вбудованих систем".</p> <ul style="list-style-type: none"> <li>– Вміти аналізувати вимоги до застосунку і підібрати компоненти апаратного та програмного забезпечення для створення ефективного застосунку.</li> <li>– Вміти вивчати документацію та технічні характеристики апаратного забезпечення та аналізувати наявні функції та потенційні обмеження.</li> <li>– Вміти створювати повноцінні застосунки на основі операційної системи реального часу з використанням зовнішніх сенсорів.</li> <li>– Вміти відлагоджувати дефекти у системах та ідентифікувати вразливості системи.</li> <li>– Вміти застосовувати та налаштовувати датчики з різними інтерфейсами та протоколами передачі даних для ефективної роботи всього застосунку</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<p>Поєднання загальних та спеціальних компетентностей, теоретичних та практичних знань, умінь та здатностей надає здобувачам ступеня бакалавр базові уявлення та досвід задля здійснення ними ефективної діяльності в сфері розробки вбудованих систем реального часу.</p>
<b>Інформаційне забезпечення</b>	<p>Силабус дисципліни, навчально-методичний комплекс.</p>
<b>Форма проведення занять</b>	<p>Лекції, практичні та семінарські заняття, лабораторні роботи</p>
<b>Семестровий контроль</b>	<p>Залік</p>
<b>Викладач</b>	<p>ас. кафедри ОТ, Гордієнко Нікіта Юрійович</p>

<b>Дисципліна</b>	<b>Технології розроблення вбудованих систем IoT (Сертифікатна програма)</b>
<b>Кафедра</b>	Обчислювальної техніки
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 90
<b>Курс, семестр</b>	4 курс, 8 семестр
<b>Обсяг</b>	4 кредити, 120 годин

<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	<p>Базові знання з дисциплін:  Дискретна математика, Комп'ютерна логіка, Комп'ютерна арифметика, Програмування C, C++, Python, Комп'ютерні мережі, Системне програмування, Архітектура комп'ютерів, Архітектура процесорів, Мікропроцесорні системи, ОС Linux.</p>
<b>Що буде вивчатися</b>	<p>Досвід розробника вбудованих систем, вже отриманий під час попереднього навчання дозволяє долучитися до розв'язання певних із означених проблем самостійно або в рамках команди і розробити вбудований пристрій із підтримкою Інтернету речей.</p> <p>Узагальнена тематика лабораторних робіт:</p> <ul style="list-style-type: none"> <li>- розробка складових елементів вбудованих систем та пристроїв IoT для рішення задач в реальному часі;</li> <li>- конфігурація та використання ОС Linux, розробка та конфігурація програмного забезпечення для мікроконтролерів на базі процесорів ARM;</li> <li>- прошивка плат,</li> <li>- розвертання ядра операційної системи Linux на процесорних ядрах ARM; розробка модулів ядра для вбудованих систем;</li> <li>- написання автотестів.</li> </ul> <p>Бакалаврські проекти:  Розроблення проекту з власною прошивкою на базі апаратних засобів для програмування та налагодження мікроконтролерів. Використання платформ BeagleBone Black, Raspberry Pi, STM StarterKit GlobalLogic, ESM32.</p>
<b>Чому це цікаво/треба вивчати</b>	<p>Вбудовані системи IoT поєднують у собі величезну кількість технологічних проблем, пов'язаних із вбудованим програмним забезпеченням, вбудованими обчисленнями, протоколами зв'язку з низьким енергоспоживанням, вбудованими протоколами безпеки, вбудованим збором енергії, обробкою даних датчиків поблизу, вбудованими методами штучного інтелекту та інтеграцією в хмарні та периферійні обчислювальні системи.</p> <p>На сьогодні застосування вбудованих систем є повсюдним та все більше поширюються, вбудовані системи наразі це невід'ємна частина Інтернету речей (IoT) та Інтернету всього (IoE). Комерційна, урядова, військова, медична, освітня, культурна та інші інфраструктури сьогодні пов'язані в єдину мережу IoT і є конгломератом, як автономних та і взаємозв'язаних вбудованих систем.</p> <p>Стрімкий розвиток сучасних технологій IoT потребує відповідної специфіки знань та вмінь від сучасних фахівців в області комп'ютерної інженерії, програмування та загалом інформаційних систем.</p> <p>Цікаві проекти інноваційних методів навчання вже реалізуються і плануються до використання в курсі «Технології розроблення вбудованих систем IoT», це – «Лабораторія по віддаленій прошивці плат». Для ініціативних студентів є купа цікавих задачок, де можна проявити себе та отримати сучасний досвід розробки та програмування для вбудованих систем. А компанія GL забезпечила сертифікатну програму базовими комплектами «заліза» для Ваших творчих досягнень.</p> <p>Курс «Технології розроблення вбудованих систем IoT» може бути</p>



	<p>прослуханий студентами як окремий курс, так і у складі <b>сертифікатної програми «Інженерія вбудованих систем та Інтернет речей»</b>, яка запроваджена на факультеті інформатики та обчислювальної техніки з 2021 – 2022 навчального року. Детально з сертифікатною програмою можна ознайомитися на сайті кафедри обчислювальної техніки <a href="https://comsys.kpi.ua/sertifikatni-programi">https://comsys.kpi.ua/sertifikatni-programi</a>.</p> <p>Перелік дисциплін сертифікатної програми:</p> <ul style="list-style-type: none"> <li>• Технології програмування C/Embedded – 5 семестр;</li> <li>• Управління IT-інфраструктурними проєктами – 6 семестр;</li> <li>• Технології програмування на ПЛІС (FPGA) – 6 семестр;</li> <li>• Тестування та контроль якості (QA) вбудованих систем – 7 семестр;</li> <li>• Технології розроблення вбудованих систем IoT – 8 семестр.</li> </ul> <p>В цьому курсі і в сертифікатній програмі загалом розглядаються цікаві задачі розроблені у співробітництві з менторами та тренерами компанії GlobalLogic, які також часто приймають участь в лекційних та практичних заняттях. Ми намагаємося допомогти студентам краще зрозуміти реалії сучасного ринку Embedded Systems та потреби роботодавців, подолати важкий поріг входження в Embedded та світ HW.</p> <p>Дефіцит фахівців о області Embedded Systems та IoT на сьогодні доволі значний, тому опанування дисциплін сертифікатної програми дозволить розширити можливості студентів в пошуку майбутньої роботи та підвищить його конкурентоспроможність. З точки зору компанії GlobalLogic участь в сертифікатній програмі дозволить сформувати необхідні знання та вміння, що задовольнять її, як роботодавця.</p>
<p><b>Чому можна навчитися (результати навчання)</b></p>	<p>Курс призначений для студентів, які люблять практичне програмування та вирішення завдань в області проєктування HW та IoT.</p> <p>Слухачі курсу узагальнять та вдосконалять практичні навички <i>програмування для мікроконтролерів:</i></p> <ul style="list-style-type: none"> <li>- програмування на мові C, C++ для мікроконтролерів STM32, ESM32;</li> <li>- розробки програмного забезпечення для МК на платі Global Logic Startet Kit STM32;</li> </ul> <p><i>Linux Kernel:</i></p> <ul style="list-style-type: none"> <li>- розвертання процесорних ядер за допомогою Toolchains.</li> <li>- розроблення програмного забезпечення в ОС Linux на платі та віртуальній машині для лінійки процесорів ARMорієнтованих на встановлення ОС;</li> </ul> <p><i>QA:</i></p> <ul style="list-style-type: none"> <li>- тестування розроблених модулів;</li> <li>- розроблення автотестів для вбудованих систем на мові Python;</li> <li>- використання системи контролю версій Git для роботи в команді;</li> <li>- генерації та аналізу мережного трафіку в інфраструктурі IoT;</li> <li>- оцінювання обчислювальних можливостей вбудованих систем IoT.</li> </ul> <p>Незважаючи на поглиблену складову HW, студенти все ж отримують основні знання та навички з різних рівнів інфраструктури IoT. Як то організація та розробка системи кінцевих датчиків та сенсорів, мережного оточення та інтернет інфраструктури для обміну даними</p>

	між пристроями IoT та серверами обробки даних, архітектури Edge та хмарної інфраструктури аналітики даних – хмар IoT.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<p>Після проходження курсу студенти зможуть розробити та тестувати програмно-технічні засоби обробки та пересилання інформації в комп'ютерних системах, що мають безпосередні зв'язки з реальними об'єктами.</p> <p>Отримані знання можна застосувати на практиці в будь якій сучасній IT компанії, особливо в компаніях, що займаються розробкою систем реального часу різного застосування:</p> <ul style="list-style-type: none"> <li>- транспортні системи для автомобілів, метро, літаків, залізниць і кораблів;</li> <li>- контроль руху автомагістралей, повітряного простору, залізничних колій і судноплавних шляхів;</li> <li>- контроль технологічних процесів для електростанцій, заводів та споживчих товарів;</li> <li>- медичні системи для терапії, моніторингу пацієнтів;</li> <li>- військове використання;</li> <li>- виробничі системи з роботами;</li> <li>- телефонний, радіо- та супутниковий зв'язок;</li> <li>- комп'ютерні ігри, мультимедійні системи, текстові, графічні, аудіо- та відеоінтерфейси;</li> <li>- побутові системи для моніторингу та керування приладами;</li> <li>- контроль об'єктів, таких як тепло, світло, двері та ліфти;</li> </ul> <p>тощо.</p>
<b>Інформаційне забезпечення</b>	Силабус дисципліни, навчально-методичний комплекс
<b>Форма проведення занять</b>	Лекції, лабораторні роботи, самостійна робота
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	Професор кафедри ОТ, д.т.н., Клименко І.А., асистент Каплунов А.В.

## Освітній компонент 14Ф-каталогу

Дисципліна	Інтелектуальні транспортні системи
Кафедра	Інформаційних систем та технологій
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	4 курс, 8 семестр
Обсяг	4 кредити, 120 годин
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з дисциплін «Комп'ютерні мережі», «Інтернет речей», «Дискретна математика», «Системна інженерія».
Що буде вивчатися	Теоретичні та практичні аспекти проектування, розробки, експлуатації сучасних інтелектуальних транспортних систем, їх основні задачі та функції.
Чому це цікаво/треба вивчати	<p>Уже понад 70 років для регулювання транспортних потоків використовуються передові технології. Виробники транспортних засобів розробляють передові технології для того, щоб транспортні засоби стали безпечнішими, пересування завдало менше стресу і було зручнішим. Передові технології все більше і більше застосовуються до великих систем громадського транспорту, а також для поширення інформації про прибуття поїздів та автобусів для пасажирів. В секторі вантажного транспорту ціла низка технологій застосовується для покращення ефективності руху транспортних засобів та відповідних комерційних операцій як ланки ланцюга постачання.</p> <p>Ці різні технології тепер відомі під збіркою назвою інтелектуальні транспортні системи (ІТС). Відтак, ІТС можна визначити як застосування комп'ютерних, інформаційних та комунікаційних технологій для управління транспортними засобами та мережами у реальному часі, включаючи переміщення людей і товарів.</p>
Чому можна навчитися (результати навчання)	<p>У результаті вивчення дисципліни студенти повинні <b>знати</b>:</p> <ul style="list-style-type: none"> <li>- характеристики інтелектуальних транспортних систем;</li> <li>- структуру інтелектуальних систем;</li> <li>- характеристики та види інтелектуальних систем для інфраструктури транспорту;</li> <li>- характеристики та види інтелектуальних систем для транспортних засобів.</li> </ul> <p><b>вміти</b>:</p> <ul style="list-style-type: none"> <li>- ідентифікувати параметри інтелектуальних транспортних систем;</li> <li>- описувати структуру інтелектуальних систем;</li> <li>- вибирати характеристики для інтелектуальних систем у сфері інфраструктури транспорту;</li> <li>- вибирати характеристики для інтелектуальних систем управління транспортними засобами.</li> </ul>
Як можна користуватися набутими знаннями і вміннями (компетентності)	Розробка та застосування елементів інтелектуальних систем на транспорті.

<b>Інформаційне забезпечення</b>	Силабус дисципліни, опорний конспект лекцій, методичні вказівки до виконання лабораторних робіт, перелік завдань для самостійної роботи студента, перелік питань до МКР та семестрового контролю.
<b>Форма проведення занять</b>	Лекції, лабораторні роботи.
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., доцент Сокульський О.Є.

<b>Дисципліна</b>	<b>Кібербезпека комп'ютерних мереж</b>
<b>Кафедра</b>	Інформатики та програмної інженерії
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	4 курс, 8 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: комп'ютерні мережі, інтернет-технології
<b>Що буде вивчатися</b>	<p>Базові поняття кібербезпеки. Різновиди шкідливого ПЗ. Інструменти вивчення обчислювальних систем та мереж. Різновиди атак. Управління ризиками. Ланцюг кібератаки. Елементи ланцюга кібератаки та засоби протидії.</p> <p>Використання стандартних мережевих утиліт для вивчення мережі. Сканування портів. nmap, zenmap. Розширені можливості nmap для вивчення мережі.</p> <p>Сканування вразливостей. Протидія скануванню вразливостей. Використання сканерів вразливостей: XSpider, Nessus, OpenVAS. Вразливості мережевих протоколів та їх використання.</p> <p>Засоби моделювання мережі. Віртуальні мережеві пристрої. Використання систем віртуалізації VMware Player/Workstation, ESXi, VirtualBox. Використання системи емуляції EVE-NG.</p> <p>Моніторинг та фільтрація трафіка. Stateless и Stateful файєрволи. Системи виявлення та запобігання вторгненням. Системи Advanced Malware Protection и Advanced Threat Protection</p> <p>Концепція системи безпеки Junos. Сучасні завдання та виклики для системи безпеки. Дизайн системи безпеки.</p> <p>Мережеві екрани Juniper SRX. Початкова конфігурація.</p> <p>Об'єкти системи безпеки. Security Zone, Security Screen, Security Address, Security Service. Створення об'єктів системи безпеки.</p> <p>Політики безпеки. Міжзонні та глобальні безпекові політики. Шлюз рівня додатків із уніфікованими політиками безпеки. Створення політик безпеки на Juniper SRX.</p> <p>Сервіси безпеки. Додавання сервісів IDP та служби ідентифікації користувачів до політик безпеки</p> <p>UTM. Контентна фільтрація. Web-фільтрація. Робота з антивірусом. Модуль Antispam.</p> <p>Хмарний сервіс SkyATP. Огляд SkyATP. Блокування загроз.</p> <p>Трансляція адрес (NAT).</p> <p>IPsec VPNs. Побудова тунелю IPsec. Обробка трафіку в тунелі IPsec. IPsec тунель "точка-точка"</p>

	Підвищення безпеки мережевих пристроїв Juniper. Вразливості локальних мереж. Підвищення безпеки локальних мереж за допомогою комутаторів Juniper.
<b>Чому це цікаво/треба вивчати</b>	Знання кібербезпеки та сучасного мережевого обладнання є суттєвою конкурентною перевагою на ринку праці. В курсі розглядаються основи кібербезпеки та використання обладнання Juniper SRX для забезпечення безпеки комп'ютерних мереж. Juniper – один з основних виробників мережевого обладнання. Обладнання Juniper широко використовується в Україні та за кордоном. Курс дозволяє засвоїти основи кібербезпеки, надбати знання та навички з конфігурування обладнання Juniper для забезпечення мережевої безпеки. Курс дозволяє підготуватись до сертифікації JNCIA-SEC.
<b>Чому можна навчитися (результати навчання)</b>	Розумінню основних понять кібербезпеки. Використанню стандартних мережевих утиліт та спеціалізованого ПЗ для вивчення мережі. Використанню мережевих екранів Juniper SRX для запобігання мережевих атак. Підготуватися до сертифікаційного іспиту JNCIA-SEC.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Розуміти основні принципи кібербезпеки. Використовувати стандартне та спеціалізоване ПЗ для вивчення мережі, пошуку та аналізу вразливостей. Виконувати конфігурування мережевих екранів Juniper SRX. Налаштувати фільтрацію трафіку та запобігання вторгнень на базі Juniper SRX. Використовувати систему Sky ATP. Налаштувати NAT. Налаштувати IPsec VPN. Використовувати комутатори Juniper для підвищення безпеки локальної мережі.
<b>Інформаційне забезпечення</b>	Навчальна і робоча програми дисципліни, електронний конспект та презентація
<b>Форма проведення занять</b>	Лекції, практичні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., Миронов Дмитро Вікторович JNCIA, JNCIS, CCAI, CCNA, CCNP

<b>Дисципліна</b>	<b>Теорія формальних граматики</b>
<b>Кафедра</b>	Інформатики та програмної інженерії
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	4 курс, 8 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Дисципліна «Теорія формальних граматики» базується на наступних дисциплінах – «Дискретна математика», «Теорія ймовірностей».

<b>Що буде вивчатися</b>	<ul style="list-style-type: none"> <li>– основи формальних граматики як формальних систем ;</li> <li>– класифікація і ієрархія формальних граматики;</li> <li>– ймовірнісні граматики, нечіткі граматики, автоматні граматики і скінченні автомати;</li> <li>абстрактні машини Тьюринга, Поста; нечіткі алгоритми, нечіткі мови, нечітка машина Тьюринга</li> </ul>
<b>Чому це цікаво/треба вивчати</b>	<ul style="list-style-type: none"> <li>• теорія формальних граматики — це шлях до аналізу існуючих мов програмування та створення нових;</li> <li>• використання класичних методів формальних граматики разом з сучасними підходами для реалізації нових мов програмування</li> </ul>
<b>Чому можна навчитися (результати навчання)</b>	<ul style="list-style-type: none"> <li>– застосовувати методи формальних граматики різних класів для опису мов, процесів, різних практичних застосувань;</li> <li>- застосовувати методи для аналізу алгоритмів, конструювання формальних систем і моделей</li> </ul>
<b>Як можна користуватися набутими знаннями і вміннями (компетентності)</b>	– застосовувати формальні граматики для аналізу і розробки програмного забезпечення конструювання програмних систем аналізувати, цілеспрямовано шукати і вибирати необхідні засоби для опису, конструювання формальних мов, формальних алгоритмів, впровадження цих підходів для формалізації процесів розв’язання складних задач
<b>Інформаційне забезпечення</b>	Навчальна та робоча програми дисципліни, РСО, лабораторні завдання
<b>Форма проведення занять</b>	Лекції та лабораторні (комп’ютерні практикуми)
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	доц., к.т.н. Баклан І.В.

<b>Дисципліна</b>	<b>Сучасні технології розробки програмного забезпечення (Авторський курс компанії SoftServe)</b>
<b>Кафедра</b>	Обчислювальної техніки
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 150
<b>Курс, семестр</b>	4 курс, 8 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: Інженерія програмного забезпечення, Програмування, Об’єктно-орієнтоване програмування, Алгоритми та методи обчислень, Системне програмування, Алгоритми і структури даних, Архітектура комп’ютерів
<b>Що буде вивчатися</b>	–основи термінології і основних понять. Основні методології розробки ПО. Методи вибору технологій для розробки. Огляд існуючих технологічних рішень які використовуються для створення проектівоснови застосування теоретичних знань на прикладі розробки документації навчального проекту персоналізованого для кожного студента чи групи студентів за їх вибором
<b>Чому це цікаво/треба вивчати</b>	–Існуючі потреби та тенденції на ринку праці показують що для

	успішної роботи програмістом потрібно не тільки вміти писати код(програмувати) також потрібне розуміння принципів розробки ПО в цілому. До 50% часу технічної співбесіди витрачається саме на питання які стосуються методологій розробки оскільки домінуючою методологією розробки ПО на даний час є AGILE отримати розуміння та практику розробки власного проекту з 0 дозволить достатньо підготуватися до майбутньої роботи
<b>Чому можна навчитися (результати навчання)</b>	Розібратися в реальному процесі розробки ПО. Підготуватися до того що буде очікувати студента після працевлаштування на реальне робоче місце.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Розуміти як читати та писати проектну документацію для проекту в цілому чи його окремих компонент. Обирати методології та технології для реалізації проекту чи його компонент.
<b>Інформаційне забезпечення</b>	Силабус дисципліни, навчально-методичний комплекс.
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	ст. викладач Шевело О.П.

<b>Дисципліна</b>	<b>Методика та організація наукових досліджень</b>
<b>Кафедра</b>	Обчислювальної техніки
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 120
<b>Курс, семестр</b>	4 курс, 7 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з дисциплін: Теорія ймовірності, Алгоритми та методи обчислень, Системне програмування, Системне програмне забезпечення, Комп'ютерні системи, Структури даних і алгоритми, Організація баз даних, Програмування, Архітектура комп'ютерів, Комп'ютерні мережі, Дискретна математика
<b>Що буде вивчатися</b>	- Методи проведення наукових досліджень - Методи математичного моделювання - Методи відображення та перетворювання інформації
<b>Чому це цікаво/треба вивчати</b>	Дисципліна закладає основи самостійної наукової роботи починаючи від лабораторних робіт, курсового та дипломного проектування
<b>Чому можна навчитися (результати навчання)</b>	Створювати різноманітні моделі, методик виконання наукових досліджень
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Планувати наукові дослідження, обробляти результати наукових досліджень, обробляти та аналізувати отримані результати
<b>Інформаційне забезпечення</b>	Силабус дисципліни, навчально-методичний комплекс.

Форма проведення занять	Лекції, лабораторні роботи
Семестровий контроль	Залік
Викладач	к.т.н., доцент Селіванов В.Л.

Дисципліна	Системи цифрової обробки сигналів і зображень
Кафедра	Обчислювальної техніки
Рівень ВО	Перший (бакалаврський)
Можливі обмеження	Кількість студентів на курс від 60 до 120
Курс, семестр	3 курс, 5 семестр
Обсяг	4 кредити ЄКТС
Мова викладання	Українська
Вимоги до початку вивчення	Базові знання з комп'ютерної логіки, комп'ютерної арифметики, дискретної математики, математичного аналізу. Базові знання архітектури комп'ютера, комп'ютерної схемотехніки.
Що буде вивчатися	<p>Методи цифрового дизайну та проектування для програмовної логічної інтегральної схеми (ПЛІС). Проектування конвеєрних пристроїв для цифрової обробки сигналів та зображень з використанням мови VHDL. Функціональне моделювання та структурний синтез в сучасних САПР.</p> <p>Вивчаються алгоритми фільтрації та швидких ортогональних перетворень одно- та двовимірних сигналів з нахилом у їхню апаратну реалізацію.</p> <p>Практичні завдання включають розробку нескладних високопродуктивних цифрових фільтрів, процесора для обробки зображень, елементів нейронної мережі. При цьому вивчається оригінальний метод, який дає змогу описувати поведінку пристрою, не будуючи та не креслячи його реальну структуру. Опис пристрою моделюється у стенді для іспитів, який призначений для систем обробки сигналів, у симуляторі фірми Aldec або Mentor. Одержаний опис компілюється компілятором-синтезатором САПР ПЛІС довільної серії, наприклад, Intel-Altera, AMD-Xilinx або Lattice у високопродуктивний конвеєрний пристрій. Цей пристрій як готовий модуль IP-ядра може бути вбудований у складний проект, зконфігурований у ПЛІС будь-якого виробника.</p> <p>Також вивчатиметься розробка застосунків для системи програмовного радіо (Software-defined radio).</p>
Чому це цікаво/треба вивчати	У даний час розробка та виготовлення спеціалізованих високопродуктивних обчислювальних засобів для вирішення задач цифрової обробки сигналів, зображень, систем штучного інтелекту можлива лише на базі ПЛІС. Також ПЛІС зараз широко використовуються для пришвидшення вирішення таких задач у датацентрах та хмарах. Отримані навички та знання будуть актуальні в областях розроблення рішень для Інтернету Речей (IoT), вбудованих систем, інтелектуальних відеосистем, систем штучного інтелекту, а також для програмування застосунків для датацентрів.
Чому можна навчитися (результати навчання)	Розробляти цифрові схеми та пристрої для обробки сигналів з використанням сучасних методів проектування та мов проектування



	<p>апаратури. Отримати навички програмування на мові VHDL. Користуватися сучасними професійними засобами для цифрового дизайну та моделювання. Виконувати функціональне моделювання і налагодження пристроїв за допомогою VHDL-симуляторів.</p> <p>Важливо, що оригінальний метод, що вивчається, дає змогу одержати пристрої, які після конфігурування у ПЛІС, як правило, є працездатними з першого разу.</p>
<p><b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b></p>	<p>Базові знання у використанні мови проектування апаратури VHDL, а також досвід і навички використання сучасних САПР ПЛІС достатньо для початку професійної діяльності в напрямку проектування апаратури для складних систем на ПЛІС та НВІС (ASIC). Також вони стануть в нагоді при програмуванні паралельних обчислювальних систем на графічних акселераторах, кристалах штучного інтелекту, вбудованих систем.</p> <p>Отримані базові знання програмування та методів цифрового дизайну дозволять за необхідності швидко перейти на використання мови програмування апаратури Verilog, мікросхем різних виробників ПЛІС (Intel-Altera, AMD-Xilinx або Lattice), а також інших САПР для функціонального та структурного синтезу на ПЛІС та НВІС (Synopsys, Aldec, Cadence).</p> <p>Отримані базові знання допоможуть у подальшому професійному навчанні в тому числі самостійно.</p>
<b>Інформаційне забезпечення</b>	Силабус дисципліни, навчально-методичний комплекс.
<b>Форма проведення занять</b>	Лекції, лабораторні роботи
<b>Семестровий контроль</b>	залік
<b>Викладач</b>	Професор, д.т.н., с.н.с Сергієнко Анатолій Михайлович

<b>Дисципліна</b>	<b>Статистичні методи машинного навчання</b>
<b>Кафедра</b>	Обчислювальної техніки
<b>Рівень ВО</b>	Перший (бакалаврський)
<b>Можливі обмеження</b>	Кількість студентів на курс від 60 до 150
<b>Курс, семестр</b>	4 курс, 8 семестр
<b>Обсяг</b>	4 кредити, 120 годин
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з вищої математики (математичний аналіз, лінійна алгебра та аналітична геометрія), теорія ймовірностей та математична статистика, програмування, навички розв'язування типових задач цих дисциплін
<b>Що буде вивчатися</b>	Основні методи машинного навчання для задач класифікації, кластеризації і регресії (прогнозування), зокрема, принципи побудови деяких основних класифікаторів, задачі, які потребують застосування статистичних методів машинного навчання, алгоритмічні та обчислювальні підходи до їх реалізації з поглибленим вивченням мови програмування Python та застосуванням бібліотек TensorFlow і THEANO.

<b>Чому це цікаво/треба вивчати</b>	<p>Для отримання істотних конкурентних переваг у сфері ІТ-технологій, оскільки машинне навчання широко використовується при побудові пошукових машин, які навчаються надавати релевантні результати; моделей для генерування таргетованої реклами в рамках масових рекламних кампаній; анти-спамного програмного забезпечення з фільтрування повідомлень електронної пошти; програмного забезпечення для виявлення шахрайства в мережі Інтернет, зокрема при транзакціях з кредитними картками; розпізнавання образів та голосових команд.</p> <p>Машинне навчання також широко використовується в наукових програмах, таких як біоінформатика, медицина та астрономія.</p>
<b>Чому можна навчитися (результати навчання)</b>	<p>Знання принципів побудови векторів ознак, правил рішень і класифікації; основних видів класифікаторів; принципів побудови лінійних класифікаторів; принципів побудови нелінійних класифікаторів; особливостей вибору ознак класифікації і попередньої обробки даних.</p> <p>Вміння вибирати відповідний вид класифікатора залежно від розв'язуваної задачі; вибирати набір ознак для класифікації і проводити попередню обробку даних; застосовувати алгоритми побудови і навчання класифікатора за вибіркою; виконувати обчислення, пов'язані з навчанням і роботою класифікатора, в середовищі Python з застосуванням бібліотек TensorFlow і THEANO.</p>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>- для вибору, побудови, навчання і використання основних класифікаторів при вирішенні завдань;</li> <li>- для самостійної роботи в сучасних програмних комплексах; освоєння великого обсягу інформації;</li> <li>- для постановки задач та проведення експерименту, програмування задач аналізу даних; для візуалізації одержаних результатів.</li> </ul>
<b>Інформаційне забезпечення</b>	Силабус дисципліни, навчально-методичний комплекс
<b>Форма проведення занять</b>	Лекції, лабораторні роботи, самостійна робота
<b>Семестровий контроль</b>	Залік
<b>Викладач</b>	к.т.н., Олещенко Л.М.