



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»  
ФАКУЛЬТЕТ ПРИКЛАДНОЇ МАТЕМАТИКИ  
КАФЕДРА ПРИКЛАДНОЇ МАТЕМАТИКИ



**ЗАТВЕРДЖЕНО**

Методичною радою  
КПІ ім. Ігоря Сікорського  
(протокол № 5 від «06» березня 2025 р.)

## **Фаховий КАТАЛОГ**

**вибіркових навчальних дисциплін  
першого (бакалаврського) рівня вищої освіти  
для освітньої програми  
«Наука про дані та математичне моделювання»  
спеціальності 113 Прикладна математика**

**УХВАЛЕНО:**

Вченою радою ФПМ  
КПІ ім. Ігоря Сікорського  
(протокол № 7 від «27» січня 2025 р.)

**Київ 2025**

# Зміст

Інструкція користувачам каталогу	3
Ф-Каталог – 2025 р.	4
<i>Дисципліни для 2 курсу (вибір першокурсників)</i>	4
<i>Дисципліни для 3 курсу (вибір другокурсників)</i>	5
<i>Дисципліни для 4 курсу (вибір третьокурсників)</i>	6
Анотації вибіркових дисциплін для 2 курсу	7
1. Об'єктно-орієнтоване програмування	7
2. Прикладна механіка	8
3. Алгоритми і технології комп'ютерної графіки і мультимедіа	9
4. Функційне програмування	10
5. Теорія функцій комплексної змінної	11
6. Front-end розробка	12
Анотації вибіркових дисциплін для 3 курсу	13
7. Дослідження операцій	13
8. Математичне програмування	14
9. Випадкові процеси	15
10. Застосування генеративного ШІ	16
11. Розподілені і хмарні обчислення	17
12. Рівняння математичної фізики	18
13. Front-end розробка	19
14. Основи геометричного моделювання	20
15. Штучний інтелект	21
Анотації вибіркових дисциплін для 4 курсу	22
16. Системи глибинного навчання	22
17. Інформаційні системи	23
18. Розподілені і хмарні обчислення	24
19. Навчання з підкріпленням у глибинному навчанні	25
20. Управління проєктами	26
21. Застосування генеративного ШІ	27
22. Geometric Modelling	28
23. Системний аналіз	29
24. Теорія оптимального керування	30
25. Методи теорії надійності та ризику	31

## Інструкція користувачам каталогу

1. Кількість і обсяг (у кредитах ЄКТС) навчальних дисциплін, які може обрати студент (вибіркових дисциплін), визначається навчальним планом, а саме: для II курсу – 8 кредитів, III курсу – 20 кредитів, IV курсу – 24 кредити. У навчальному плані зазначається також семестр, у якому викладають вибірку дисципліну, форма семестрового контролю, види та обсяги навчальних занять.

2. Безпосередній вибір студентами дисциплін здійснюється через сайт <https://my.kpi.ua/>.

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

4. Здобувача, який знехтував своїм правом вибору, може бути записано на вивчення навчальних дисциплін, які обрав завідувач випускової кафедри, для оптимізації навчальних груп/потоків.

5. Якщо здобувач із поважної причини не зміг обрати дисципліни вчасно, або виявив помилку щодо свого волевиявлення, він звертається до деканату з заявою для запису на вивчення обраних ним дисциплін, надавши документи, що засвідчують поважність причин. Заява на зміну вибіркової дисципліни у сформованому індивідуальному навчальному плані має подаватися не пізніше, ніж за місяць до початку семестру, в якому викладають цю дисципліну.

6. Не допускається зміна обраних дисциплін після початку навчального семестру, в якому їх викладають.

7. Навчальні дисципліни, які обрав студент, зазначаються в його індивідуальному навчальному плані.

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

## Ф-Каталог – 2025 р.

*Дисципліни для 2 курсу (вибір першокурсників)*  
**(потрібно обрати 8 кредитів)**

**3-й семестр: одна дисципліна обсягом у 4 кредити**

**4-й семестр: одна дисципліна обсягом у 4 кредити**

№	Назва навчальної дисципліни	Семестр	Кіл-ть кредитів	Семестрова атестація
1	Об'єктно-орієнтоване програмування	3	4	залік
2	Прикладна механіка	3	4	залік
3	Алгоритми і технології комп'ютерної графіки і мультимедіа	3	4	залік
4	Функційне програмування	4	4	залік
5	Теорія функцій комплексної змінної	4	4	залік
6	Front-end розробка	4	4	залік

*Дисципліни для 3 курсу (вибір другокурсників)*

**(потрібно обрати 20 кредитів)**

**5-й семестр: три дисципліни обсягом у 4 кредити кожна**

**6-й семестр: дві дисципліни обсягом у 4 кредити кожна**

<b>№</b>	<b>Назва навчальної дисципліни</b>	<b>Семестр</b>	<b>Кіл-ть кредитів</b>	<b>Семестрова атестація</b>
1	Дослідження операцій	5	4	залік
2	Математичне програмування	5	4	залік
3	Випадкові процеси	5	4	залік
4	Застосування генеративного ШІ	5	4	залік
5	Розподілені і хмарні обчислення	5	4	залік
6	Рівняння математичної фізики	6	4	залік
7	Front-end розробка	6	4	залік
8	Основи геометричного моделювання	6	4	залік
9	Штучний інтелект	6	4	залік

*Дисципліни для 4 курсу (вибір третьокурсників)*

**(потрібно обрати 24 кредити)**

**7-й семестр: чотири дисципліни обсягом у 4 кредити кожна**

**8-й семестр: дві дисципліни обсягом у 4 кредити кожна**

№	Назва навчальної дисципліни	Семестр	Кіл-ть кредитів	Семестрова атестація
1	Системи глибинного навчання	7	4	залік
2	Інформаційні системи	7	4	залік
3	Розподілені і хмарні обчислення	7	4	залік
4	Навчання з підкріпленням в глибинному навчанні	7	4	залік
5	Управління проектами	7	4	залік
6	Застосування генеративного ШІ	7	4	залік
7	Geometric Modelling	7	4	залік
8	Системний аналіз	8	4	залік
9	Теорія оптимального керування	8	4	залік
10	Методи теорії надійності та ризику	8	4	залік

## Анотації вибірових дисциплін для 2 курсу

<b>1. Об'єктно-орієнтоване програмування</b>	
<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	2, 3
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення дисципліни</b>	Розуміння принципів процедурного програмування мовою С. Розуміння класичних алгоритмів та структур даних.
<b>Що буде вивчатися</b>	Головні принципи об'єктно-орієнтованої розробки, SOLID, реалізація багат шарової архітектури.
<b>Чому це цікаво/треба вивчати</b>	ООП — найбільш поширена парадигма програмування у сучасному світі, дана парадигма застосовується при реалізації широкого спектру програм, таких як користувацькі інтерфейси, серверна логіка web-додатків чи математичні моделі в Data Science.
<b>Чому можна навчитися</b>	Ідентифікувати та реалізувати програмно об'єктне представлення предметної області. Ідентифікувати недоліки у об'єктному представленні предметної області з використанням принципів SOLID, змінювати їх через рефакторинг. Ідентифікувати та реалізовувати найбільш поширені шаблони (патерни) проектування. Розуміти внутрішню структуру представлення об'єктів у мові Java. Розробляти структуру проекту через діаграму класів UML та відповідно реалізовувати програмні продукти мовою Java. Розробляти програмний продукт з гарантією якості через створені до нього тести. Бачити код очима замовника, тобто через постановку задачі та її реалізацію користуючись принципами BDD та TDD.
<b>Як можна користуватися набутими знаннями і уміннями</b>	По завершенню курсу, студенти будуть в змозі використовувати свої знання для реалізації проектів мовою Java довільної складності. Фундаментальні знання щодо принципів об'єктно-орієнтованої розробки можуть бути використані для розуміння інших мов програмування. Студентам привчається розуміння доречного використання засобів ШІ, спрямоване не на написання коду замість студента, а на виправлення його помилок.
<b>Інформаційне забезпечення дисципліни</b>	Монографії та онлайн-підручники. Відеолекції. Репозиторій лекційних прикладів і тестових завдань. Онлайн-курси за методикою змішаного навчання.
<b>Вид семестрового контролю</b>	Залік

<b>2. Прикладна механіка</b>	
Кафедра, яка забезпечує викладання	прикладної математики
Рівень вищої освіти	Перший (бакалаврський)
Курс, семестр	2, 3
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Базові знання з дисциплін «Алгебра та геометрія», «Математичний аналіз», «Основи класичної фізики».
Що буде вивчатися	<p><i>Статика твердого тіла</i> Плоска система сил. Просторова система сил. <i>Кінематика</i> Рух точки. Найпростіші види руху твердого тіла. <i>Динаміка</i> <i>Пряма та обернена задача динаміки матеріальної точки.</i> <i>Динаміка плоского руху твердого тіла.</i></p>
Чому це цікаво/треба вивчати	Термін «механіка» в буквальному перекладі з грецької означає «хитромудрість». Перед сучасними вченими постають великі задачі: засвоєння космосу, автоматика, сучасна технологія тощо. У другій половині ХХ ст. з'явився новий напрям науки – робототехніка, основою якого стала теоретична механіка. Дисципліна «Прикладна механіка» закладає теоретичні основи для можливості математичного моделювання різноманітних механічних явищ.
Чому можна навчитися	У разі успішного завершення курсу студент повинен <u>знати</u> : методику визначення сил, що діють в механічних системах тіл; умови рівноваги твердого тіла та механічної систем тіл; методи кінематичного аналізу різноманітних видів механічного руху; способи завдання руху твердого тіла; найпростіші види руху твердого тіла; <u>вміти</u> : розв'язувати задачі про визначення рівноваги тіла під дією плоскої та просторової системи сил, перевіряти правильність розрахунків; визначати основні кінематичні характеристики руху тіла в цілому та його окремих точок у випадку його найпростішого руху; розв'язувати основні задачі динаміки точки; будувати диференціальні рівняння руху точки.
Як можна користуватися набутими знаннями і уміннями	Формулювати та наводити математичну постановку задач щодо визначення умов рівноваги, критичного навантаження, внутрішніх та зовнішніх зусиль механічної конструкції під дією плоскої та просторової системи сил. Отримувати розв'язки задач управління на прикладі простих механічних систем.
Інформаційне забезпечення дисципліни	І. А. Костюшко <i>Теоретична механіка</i> Навчальний посібник для здобувачів ступеня вищої освіти бакалавра спеціальності «Прикладна математика» освітньо-професійної програми «Прикладна математика» - Запоріжжя: ЗНУ, 2019. - 222 с. Костюшко І.А., Любашенко Н.Д. «Прикладна механіка. Лабораторний практикум» 2023 р. <a href="https://ela.kpi.ua/handle/123456789/57254">https://ela.kpi.ua/handle/123456789/57254</a> Силабус. Онлайн-підручники.
Вид семестрового контролю	Залік



<b>3. Алгоритми і технології комп'ютерної графіки і мультимедіа</b>	
<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	2, 3
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення дисципліни</b>	Лінійна алгебра, математичний аналіз, алгоритми і структури даних, програмування
<b>Що буде вивчатися</b>	Математичні та алгоритмічні основи технологій комп'ютерної графіки та інших засобів мультимедіа. Алгоритми стиснення графічної та іншої інформації. Особливості застосування засобів роботи з графікою, графічних форматів. Основи обчислювальної геометрії. Фрактальна геометрія. Технології створення та відображення відеопотоків. Слухачам пропонується на вибір три цикли лабораторних робіт по напрямкам графічний дизайн, програмування засобів відображень даних, Робота з API, САПР, VR/AR контентом.
<b>Чому це цікаво/треба вивчати</b>	На сьогоднішній день стрімко змінюються технології відтворення зображень, відео та іншого мультимедійного контенту. Тому спеціалісти, які володіють такими навичками мають великий попит на ринку праці в IT індустрії. Вміння розбиратися в сучасних технологіях, які засновані на раніше вивченому математичному апараті є вельме корисною навичкою для подальшого кар'єрного росту.
<b>Чому можна навчитися</b>	Кваліфіковано використовувати сучасні засоби відображення графічної інформації. Розбиратися в технологічному ланцюжку обробки зображень. Працювати з програмним забезпеченням для обробки мультимедійного контенту від прикладних програм до програмних бібліотек.
<b>Як можна користуватися набутими знаннями і уміннями</b>	Розуміння технологічного ланцюжка і особливостей створення передачі та обробки зображень відкриває нові можливості для професійного розв'язування задач комп'ютерного зору, процесів розпізнавання та обробки зображень.
<b>Інформаційне забезпечення дисципліни</b>	Матеріали дисципліни повністю викладені на платформі Сікорський та у вигляді посилань на перевірені відкриті сучасні навчальні ресурси.
<b>Вид семестрового контролю</b>	Залік

<b>4. Функційне програмування</b>	
<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	2, 4
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення дисципліни</b>	Навички впевненого програмування на Python чи будь-якій іншій імперативній чи об'єктно-орієнтованій мові. Розуміння принципів імперативного програмування.
<b>Що буде вивчатися</b>	Функційний підхід до програмування мовою Haskell; алгебраїчні типи даних та рекурсивні типи; монади та railway-oriented programming; принципи автоматичного доведення теорем.
<b>Чому це цікаво/треба вивчати</b>	Функційне програмування — парадигма програмування, що максимально близька до математики та стрімко набирає популярності у світі. Завдяки гнучкій та потужній системі типів, що дозволяє максимально точно представляти предметну область та ефективним методам маніпуляції даними, що побудовані задля уникнення виникненню проблем у програмі, функційне програмування дозволяє доводити правильність роботи програм під час їх компіляції, дає можливість програмного опису математичних тверджень та автоматичного доведення теорем. Haskell є найбільш характерним представником чистих функційних мов.
<b>Чому можна навчитися</b>	Розробляти програми мовою Haskell, використовуючи усі переваги її системи типів та статичного аналізатору коду. Ідентифікувати проблеми в імперативному коді та застосовувати функційні підходи до їх вирішення. Розуміти принципи роботи компіляторів та систем перевірки типів. Розуміти принципи автоматичного доведення теорем.
<b>Як можна користуватися набутими знаннями і уміннями</b>	По закінченню курсу, студенти будуть у змозі використовувати функційну парадигму програмування та розробляти програмні засоби використовуючи мову Haskell, з акцентом на застосування багатої системи типів, відсутньої в інших популярних мовах програмування.
<b>Інформаційне забезпечення дисципліни</b>	Онлайн-підручники. Монографії з функційного програмування. Репозиторій лекційних прикладів. Онлайн-курси за методикою змішаного навчання. Силабус
<b>Вид семестрового контролю</b>	Залік

<b>5. Теорія функцій комплексної змінної</b>	
Кафедра, яка забезпечує викладання	прикладної математики
Рівень вищої освіти	Перший (бакалаврський)
Курс, семестр	2, 4
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Базові знання з дисциплін «Алгебра та геометрія», «Математичний аналіз».
Що буде вивчатися	Основні поняття та методи диференціального та інтегрального числення функції комплексної змінної, теорії рядів комплексного аргументу, теорії конформних перетворень, операційного числення.
Чому це цікаво/треба вивчати	Теорія функцій комплексної змінної має численні застосування в теорії пружності, термодинаміці, електротехніці і радіотехніці тощо. Апарат комплексного аналізу придатний для дослідження та обчислення дійсних інтегралів, рядів, рівнянь, а також для розв'язування багатьох інженерних задач. Ефективним при розв'язуванні диференціальних рівнянь, в тому числі з частинними похідними, є застосування операційного числення - одного з методів теорії функцій комплексної змінної
Чому можна навчитися	У разі успішного завершення курсу студент повинен <u>знати</u> : умови диференційованості функцій комплексної змінної; підходи до інтегрування функцій комплексної змінної; методи теорії лишків; операційне числення; <u>вміти</u> : диференціювати функції комплексної змінної; інтегрувати функції комплексної змінної в комплексній площині, в т. ч. застосовувати теорію лишків до обчислення деяких визначених інтегралів від функції дійсної змінної; розкласти функції комплексної змінної в ряд Лорана; знаходити конформні відображення стандартних областей; застосовувати перетворення Лапласа до розв'язування диференціальних рівнянь із сталими коефіцієнтами та їх систем
Як можна користуватися набутими знаннями і уміннями	Теоретичні положення та методи теорії функцій комплексної змінної знаходять важливі застосування при розв'язуванні прикладних задач теорії автоматичного керування, електротехніки, радіотехніки, теоретичної механіки тощо.
Інформаційне забезпечення дисципліни	Класичні підручники та монографії. Силабус.
Вид семестрового контролю	Залік

<b>6. Front-end розробка</b>	
<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	2, 4
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення дисципліни</b>	Розуміння принципів об'єктно-орієнтованої розробки. Розуміння однієї з C-подібних об'єктно-орієнтованих мов (C\C++, Java, C#). Розуміння принципів роботи комп'ютерних мереж.
<b>Що буде вивчатися</b>	Протокол HTTP. Розробка клієнт-орієнтованого коду мовою JavaScript для виконання у браузері. Технологія AJAX. Мови розмітки HTML та CSS. Платформа Node.js. Фреймворк React.
<b>Чому це цікаво/треба вивчати</b>	Веброботка — одна з найбільших частин нинішнього ринку застосунків для користувачів. Кожен має браузер і доступ в інтернет майже кожної хвилини, відтак може запускати веб-застосунки та сайти, тому вони стають все більш функціональними. В цьому курсі ми розглянемо основи роботи з клієнтською частиною вебзастосунків — від верстки вебсторінок до програмування реактивної поведінки на дії користувача, користуючись популярним фреймворком React від Facebook.
<b>Чому можна навчитися</b>	Розуміти принципи роботи протоколу HTTP. Розробляти дизайн вебсторінок за допомогою мов HTML та CSS. Розробляти динамічну поведінку вебсторінок за допомогою мови JavaScript. Користуватися платформою Node.js для запуску JavaScript коду локально. Користуватися фреймворком React та спеціальним діалектом JSX.
<b>Як можна користуватися набутими знаннями і уміннями</b>	По завершенню курсу, студенти будуть у змозі реалізувати клієнтську частину вебзастосунку: від верстки до динамічної взаємодії із сервером, за допомогою традиційних засобів HTML/CSS/JavaScript та фреймворк React.
<b>Інформаційне забезпечення дисципліни</b>	Слайди презентацій та онлайн-репозиторій лекційних прикладів. Сайти документації розглянутого програмного забезпечення. Онлайн-курси за методикою змішаного навчання. Силабус.
<b>Вид семестрового контролю</b>	Залік

## Анотації вибірових дисциплін для 3 курсу

<b>7. Дослідження операцій</b>	
Кафедра, яка забезпечує викладання	прикладної математики
Рівень вищої освіти	Перший (бакалаврський)
Курс, семестр	3, 5
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Базові знання з дисциплін «Алгебра та геометрія», «Математичний аналіз», «Програмування», «Алгоритми і структури даних»
Що буде вивчатися	Основні поняття та методи дослідження операцій. Особливу увагу приділено розділу дослідження операцій – теорії ігор - теорії прийняття оптимальних рішень в умовах конфлікту, де успіх окремого гравця, що робить вибір, залежить від вибору інших учасників.
Чому це цікаво/треба вивчати	Теорія ігор займається дослідженням математичних моделей конфліктів та їх формальним вирішенням, що дозволяє: змоделювати процес та можливі результати майбутньої гри ще до її фактичного початку; за результатами моделювання майбутньої гри прийняти рішення щодо доцільності участі та оптимальної поведінки в реальному конфлікті.
Чому можна навчитися	У разі успішного завершення курсу студент повинен <u>знати</u> : основні поняття та класифікацію видів ігор; формулювати задачу в термінах теорії ігор; методи знаходження оптимальних змішаних стратегій в антагоністичних та біматричних іграх; <u>вміти</u> : будувати матриці виграшів; отримувати розв'язки задачі в чистих та змішаних стратегіях; розв'язувати матричну / біматричну гру графічним методом; визначати розв'язок матричної / біматричної гри із використання ПЗ шляхом її приведення до задач лінійного/змішано-цілочисельного програмування.
Як можна користуватися набутими знаннями і уміннями	Моделі теорії ігор в нинішній час знаходять застосування в таких різноманітних областях, що важко усі перерахувати. В економіці для управління бізнесом, фінансами, інвестиціями, для оцінки ефективності та управління проектами, в торгівлі, маркетингу, страхуванні, в управлінні містом і т.п. Також теорія ігор використовується в екологічній області, військовій справі і в області забезпечення правопорядку, в багатьох інших областях.
Інформаційне забезпечення дисципліни	Слайди презентацій, класичні підручники, методичні посібники, онлайн підручник Силабус
Вид семестрового контролю	Залік

<b>8. Математичне програмування</b>	
<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	3, 5
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення дисципліни</b>	Базові знання з дисциплін «Алгебра та геометрія», «Математичний аналіз», «Програмування», «Алгоритми і структури даних»
<b>Що буде вивчатися</b>	Класи задач, які розглядає математичне програмування. Основні поняття математичного програмування. Методи та алгоритми оптимізації для розв'язання задач лінійного, дискретного та нелінійного програмування.
<b>Чому це цікаво/треба вивчати</b>	Математичне програмування (МП) – це інструмент, який поєднує математику, алгоритми, аналіз даних та оптимізацію, дозволяючи знаходити найкращі ефективні рішення при розв'язанні складних реальних задач для аналітиків, математиків, програмістів і дослідників у сфері оптимізації. 1. МП допомагає вирішувати реальні проблеми: <ul style="list-style-type: none"> <li>- оптимізувати логістичні маршрути, щоб зменшити витрати на доставку,</li> <li>- планувати фінанси та інвестиції для отримання максимального прибутку,</li> <li>- автоматизувати виробництво та розподіл ресурсів,</li> <li>- оптимізувати рекламні кампанії та ефективно розподіляти бюджет та ін.</li> </ul> 2. МП навчає структурованому підходу до розв'язання складних задач. Допомагає логічно аналізувати проблеми та знаходити найкращі рішення, які можуть оптимізувати процеси. 3. МП є універсальним та використовується в багатьох галузях : <ul style="list-style-type: none"> <li>- бізнес-аналітика – покращення стратегій компаній,</li> <li>- логістика та транспорт – планування маршрутів, мінімізація витрат,</li> <li>- фінанси та банківська справа – управління активами, кредитний скоринг,</li> <li>- інженерія та виробництво – розподіл ресурсів, автоматизація виробничих процесів,</li> <li>- державне управління – оптимізація розподілу бюджетів, міське планування, та ін.</li> </ul>
<b>Чому можна навчитися</b>	<ul style="list-style-type: none"> <li>- будувати оптимізаційні математичні моделі, які описують реальні проблеми,</li> <li>- логічно структурувати задачі та вибирати найкращі методи їх розв'язку,</li> <li>- оптимізувати алгоритми та зменшувати обчислювальну складність,</li> <li>- знаходити оптимальні рішення із мінімальними витратами чи максимальним прибутком.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями</b>	Використовувати набуті теоретичні знання та практичні навички при побудові та дослідженні математичних моделей обчислювальних та інформаційних процесів, пов'язаних з функціонуванням об'єктів професійної діяльності з використанням комп'ютерної техніки. Наприклад при вирішенні багаточисельних проблем управління і планування виробничих процесів, в задачах проектування та перспективного планування, при організації функціонування та розвитку соціальних процесів, їх координації з господарськими та економічними процесами.
<b>Інформаційне забезпечення дисципліни</b>	Слайди презентацій, класичні підручники, методичні посібники, онлайн підручник, силабус
<b>Вид семестрового контролю</b>	Залік

<b>9. Випадкові процеси</b>	
<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	3, 5
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення дисципліни</b>	Базові знання з дисциплін «Математичний аналіз», «Диференціальні рівняння», «Теорія ймовірностей», «Математична статистика», «Чисельні методи»
<b>Що буде вивчатися</b>	Основи теорії випадкових процесів, Марківські випадкові процеси, ланцюги Маркова, аналітичні властивості випадкових процесів, перетворення випадкових процесів, стаціонарні випадкові процеси
<b>Чому це цікаво/треба вивчати</b>	Теорія випадкових процесів лежить в основі стохастичних моделей та алгоритмів, що використовуються в аналізі даних, зокрема й інтелектуальному аналізі даних
<b>Чому можна навчитися</b>	Використовувати випадкові процеси у практичних застосуваннях, зокрема, інженерній практиці, аналізі даних та інтелектуальному аналізі даних. Розуміти принципи роботи з стохастичними моделями даних. Реалізувати аналіз даних з використанням бібліотек сучасних програмних засобів.
<b>Як можна користуватися набутими знаннями і уміннями</b>	По закінченні курсу студенти будуть у змозі використовувати основи випадкових процесів при розв'язанні прикладних задач (в т. ч. для аналізу даних) та застосовувати програмні засоби для їх вирішення.
<b>Інформаційне забезпечення дисципліни</b>	Монографії та онлайн-підручники. Відеолекції. Онлайн-курси за методикою змішаного навчання. Силабус.
<b>Вид семестрового контролю</b>	залік

<b>10. Застосування генеративного ШІ</b>	
<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	3, 5
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення дисципліни</b>	Знання з дисциплін «Архітектура обчислювальних систем», «Програмування», «Алгоритми і структури даних»
<b>Що буде вивчатися</b>	Основи генеративного штучного інтелекту, архітектури нейронних мереж, які використовуються для генерації контенту, методи навчання генеративних моделей, а також практичне застосування генеративного ШІ для створення тексту, зображень, відео та інших типів даних.
<b>Чому це цікаво/треба вивчати</b>	Генеративний штучний інтелект є однією з найдинамічніших технологій сучасності, що впливає на такі галузі, як мистецтво, дизайн, програмування, наука та бізнес-аналітика. Його вивчення дозволяє зрозуміти, як моделі ШІ можуть створювати тексти, зображення, музику, код програм та навіть креативні рішення для складних завдань. Завдяки цьому курсу студенти отримають знання про роботу передових нейромереж, таких як GPT, DALL·E та ін., навчатися розробляти та адаптувати генеративні моделі під реальні потреби. Вивчення цієї дисципліни надає конкурентні переваги на ринку праці, оскільки технології генеративного ШІ вже активно впроваджуються у великих технологічних компаніях, стартапах та дослідницьких центрах.
<b>Чому можна навчитися</b>	<ul style="list-style-type: none"> <li>- принципи роботи генеративних моделей, таких як автокодер (VAE), генеративно-змагальні мережі (GAN) та трансформери (GPT, BERT);</li> <li>- методи навчання генеративних нейронних мереж та їх оптимізацію;</li> <li>- особливості генерації тексту, зображень, аудіо та відео на основі ШІ;</li> <li>- застосування ШІ для оптимізації та вдосконалення коду програм;</li> <li>- застосування генеративного ШІ для автоматизації творчих процесів;</li> <li>- технічні та етичні аспекти використання генеративного ШІ.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями</b>	<ul style="list-style-type: none"> <li>- розробляти та навчати генеративні моделі для створення нових даних;</li> <li>- аналізувати та вдосконалювати моделі ШІ для підвищення їх ефективності;</li> <li>- застосовувати генеративний ШІ для створення контенту, автоматизації бізнес-завдань та вирішення наукових проблем;</li> <li>- інтегрувати генеративний ШІ у веб-додатки та мобільні сервіси;</li> <li>- розуміти ризики, пов'язані з використанням генеративних моделей, та застосовувати етичні стандарти.</li> </ul>
<b>Інформаційне забезпечення дисципліни</b>	Слайди презентацій, класичні підручники, методичні посібники, онлайн підручник. Силабус
<b>Вид семестрового контролю</b>	Залік



<b>11.Розподілені і хмарні обчислення</b>	
<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	3, 5
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення дисципліни</b>	Знання з дисциплін «Архітектура обчислювальних систем», «Програмування», «Алгоритми і структури даних»
<b>Що буде вивчатися</b>	Основні поняття та концепції розподілених і хмарних обчислень, методів та підходів до розв'язання прикладних задач за допомогою розподілених і хмарних обчислень.
<b>Чому це цікаво/треба вивчати</b>	Завдяки об'єднанню ресурсів і непостійного характеру споживання з боку споживачів, розподілені і хмарні обчислення дозволяють економити на масштабах, використовуючи менші апаратні ресурси, ніж були потрібні б при виділених апаратних потужностях для кожного користувача.
<b>Чому можна навчитися</b>	<ul style="list-style-type: none"> <li>- мета побудови розподілених і хмарних систем;</li> <li>- поняття розподілених обчислень та розподіленої системи, хмарних обчислень та хмарних систем;</li> <li>- вимоги до розподілених і хмарних систем;</li> <li>- класифікація розподілених і хмарних систем та їх призначення;</li> <li>- переваги та недоліки розподілених і хмарних обчислень;</li> <li>- основні парадигми застосування розподілених і хмарних обчислень;</li> <li>- моделі розподілених і хмарних архітектур.</li> <li>- паралельні чисельні алгоритми для вирішення типових задач обчислювальної математики.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями</b>	<ul style="list-style-type: none"> <li>- розробляти моделі розподілених обчислень;</li> <li>- проводити аналіз комунікаційної трудомісткості розподілених алгоритмів;</li> <li>- аналізувати та обирати оптимальні рішення щодо залучення засобів розподілених і хмарних обчислень у напрямку їх застосування для проведення наукових досліджень, а також для створення єдиного обчислювального середовища організації, підприємства;</li> <li>- застосовувати базові знання стандартів в області інформаційних технологій під час розробки та впровадження розподілених обчислювальних систем на базі хмарних технологій та сервісів;</li> <li>- проектувати компоненти програмного забезпечення для роботи в якості сервісів у складі розподілених обчислювальних систем та комплексів й хмарних обчислень.</li> </ul>
<b>Інформаційне забезпечення дисципліни</b>	Слайди презентацій, класичні підручники, методичні посібники, онлайн підручник. Силабус
<b>Вид семестрового контролю</b>	Залік

<b>12. Рівняння математичної фізики</b>	
<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	3, 6
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
<b>Мова викладання</b>	українська
<b>Вимоги до початку вивчення дисципліни</b>	Базові знання з дисциплін «Математичний аналіз», «Диференціальні рівняння», «Алгебра та геометрія»
<b>Що буде вивчатися</b>	Лінійні диференціальні рівняння в частинних похідних. Задачі Коші та початково-крайові задачі для рівнянь в частинних похідних. Методи розв'язування задач для рівнянь в частинних похідних.
<b>Чому це цікаво/треба вивчати</b>	Математична фізика розвивалася паралельно розвитку математики і фізики, її предметна область — це вивчення диференціальних рівнянь в частинних похідних, які є математичними моделями задач, що виникають в багатьох прикладних науках. Зокрема, такі рівняння описують процеси коливальних, дифузії у неперервних середовищах, явища теплопровідності тощо. Апарат математичної фізики є основним способом дослідження фізичних полів та процесів в неперервних середовищах і широко застосовується в обчислювальному експерименті та математичному моделюванні.
<b>Чому можна навчитися</b>	Будувати математичні моделі фізичних явищ, визначати тип математичної моделі (класифікувати рівняння) та обирати метод аналітичного розв'язання, розв'язувати задачі Коші для хвильових рівнянь та рівнянь теплопровідності, розв'язувати крайові задачі для еліптичних, гіперболічних та параболічних рівнянь.
<b>Як можна користуватися набутими знаннями і уміннями</b>	По закінченню курсу студенти зможуть проводити аналіз математичних моделей різноманітних процесів, що описуються лінійними рівняннями в частинних похідних, будувати розв'язки відповідних крайових задач, проводити аналіз отриманих результатів.
<b>Інформаційне забезпечення дисципліни</b>	Онлайн-підручники, тестові завдання. Силабус
<b>Вид семестрового контролю</b>	Залік

<b>13. Front-end розробка</b>	
<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	3, 6
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЕКТС: 54 аудиторні години, 66 годин самостійної роботи
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення дисципліни</b>	Розуміння принципів об'єктно-орієнтованої розробки. Розуміння однієї з C-подібних об'єктно-орієнтованих мов (C\C++, Java, C#). Розуміння принципів роботи комп'ютерних мереж.
<b>Що буде вивчатися</b>	Протокол HTTP. Розробка клієнт-орієнтованого коду мовою JavaScript для виконання у браузері. Технологія AJAX. Мови розмітки HTML та CSS. Платформа Node.js. Фреймворк React.
<b>Чому це цікаво/треба вивчати</b>	Веброзробка — одна з найбільших частин нинішнього ринку застосунків для користувачів. Кожен має браузер і доступ в інтернет майже кожної хвилини, відтак може запускати веб-застосунки та сайти, тому вони стають все більш функціональними. В цьому курсі ми розглянемо основи роботи з клієнтською частиною вебзастосунків — від верстки вебсторінок до програмування реактивної поведінки на дії користувача, користуючись популярним фреймворком React від Facebook.
<b>Чому можна навчитися</b>	Розуміти принципи роботи протоколу HTTP. Розробляти дизайн вебсторінок за допомогою мов HTML та CSS. Розробляти динамічну поведінку вебсторінок за допомогою мови JavaScript. Користуватися платформою Node.js для запуску JavaScript коду локально. Користуватися фреймворком React та спеціальним діалектом JSX.
<b>Як можна користуватися набутими знаннями і уміннями</b>	По завершенню курсу, студенти будуть у змозі реалізувати клієнтську частину вебзастосунку: від верстки до динамічної взаємодії із сервером, за допомогою традиційних засобів HTML/CSS/JavaScript та фреймворк React.
<b>Інформаційне забезпечення дисципліни</b>	Слайди презентацій та онлайн-репозиторій лекційних прикладів. Сайти документації розглянутого програмного забезпечення. Онлайн-курси за методикою змішаного навчання. Силабус.
<b>Вид семестрового контролю</b>	Залік

<b>14. Основи геометричного моделювання</b>	
<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	3, 6
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення дисципліни</b>	Математичний аналіз, чисельні методи, лінійна алгебра
<b>Що буде вивчатися</b>	Крива на площині та у просторі. Характеристики кривих (кривизна, скрут). Побудова кривих по їх характеристиках, деформування та повороти кривих. Сплайни Безьє, B-сплайни, NURBS, коротаційні балкові сплайни, сплайни на основі ділянок кіл та клотоїд.
<b>Чому це цікаво/треба вивчати</b>	Задачі геометричного дизайну, інтерполяції та згладжування. Тензорні поверхневі сплайни Безьє, B-сплайни, сплайни отримані як рішення диференційних задач математичної фізики.
<b>Чому можна навчитися</b>	Математичне визначення естетичності кривої чи поверхні і відповідні методи покращання форми фігури чи поверхні.
<b>Як можна користуватися набутими знаннями і уміннями</b>	Застосування сплайнів і сплайнових кернелів для задач обробки статистичних даних.
<b>Інформаційне забезпечення дисципліни</b>	Вся комп'ютерна графіка – це сплайни. Сплайни застосовуються в програмах анімації; для дизайну машин, приборів, фігур. Сплайни мають велике значення для розпізнавання образів. Всі програми реверс-інжиніринга базуються на сплайнах; сплайни потрібні для обробки результатів сканування лазерними сканерами чи радарами. Для управління автономними наземними, повітряними та підводними дронами також потрібні сплайни, зокрема для прокладання глобального шляху та його локального уточнення з допомогою чи то комп'ютерного зору чи то GPS даних або інерційних вимірювальних систем (IMU).
<b>Вид семестрового контролю</b>	Залік

<b>15. Штучний інтелект</b>	
<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	3, 6
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення дисципліни</b>	Базові знання з наступних дисциплін: Математичний аналіз, Алгебра і геометрія, Математична логіка, Теорія ймовірності, Програмування на мові Python.
<b>Що буде вивчатися</b>	Базові принципи ймовірнісних моделей в умовах невизначеності середовища Основи Байєсівських мереж Основи прийняття рішень в умовах невизначеності Базові методи машинного навчання (лінійна регресія, k-NN, дерева рішень, K-Means, DBSCAN) Основи нейронних мереж Основи навчання з підкріпленням Вступ до обробки природної мови Основи комп'ютерного зору
<b>Чому це цікаво/треба вивчати</b>	Базові знання зі Штучного Інтелекту допоможуть розуміти сучасні технології у сфері Машинного Навчання та Аналізу Даних. Вивчення ймовірнісних моделей дає змогу аналізувати дані та робити базові прогнози. Знання з машинного навчання дозволять працювати з даними у розрізі вирішення задач класифікації, прогнозування, кластеризації. Обробка мови та зображень є основою для створення чат-ботів, асистентів та систем розпізнавання.
<b>Чому можна навчитися</b>	Розуміти принципи, базові методи і технології у сфері Штучного Інтелекту Будувати прості моделі машинного навчання. Застосовувати базові алгоритми аналізу даних. Розбиратися в основних підходах до отримання знань в умовах невизначеності, обробки тексту та зображень.
<b>Як можна користуватися набутими знаннями і уміннями</b>	Отримані базові знання з ШІ допоможуть краще розуміти сучасні технології, дозволять реалізовувати прості проекти з аналізу даних, машинного навчання, а також слугуватимуть фундаментом для подальшого навчання у цій сфері.
<b>Інформаційне забезпечення дисципліни</b>	Слайди презентацій, монографії, наукові статті, онлайн підручники, силабус, матеріали для виконання лабораторних робіт, тестові завдання
<b>Вид семестрового контролю</b>	Залік

## Анотації вибірових дисциплін для 4 курсу

<b>16. Системи глибинного навчання</b>	
<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	4, 7
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення дисципліни</b>	Базові знання з дисциплін: «Математична логіка та теорія алгоритмів», «Алгоритми і структури даних», «Математична статистика», «Аналіз даних», «Архітектура обчислювальних систем», «Програмування»
<b>Що буде вивчатися</b>	Системи глибинного навчання (Deep Learning) - це міждисциплінарна прикладна галузь, що вивчає системи і процеси, які призначені для аналізу і обробки даних за допомогою найбільш сучасних типів глибоких нейронних мереж
<b>Чому це цікаво/треба вивчати</b>	Deep Learning, це сучасна тенденція розвитку методів і засобів обробки і аналізу даних із застосуванням глибоких нейронних мереж. Оволодіння інструментами створення і використання глибоких нейронних мереж є невід'ємним елементом підготовки висококваліфікованих спеціалістів у різних галузях національного господарства України та світу
<b>Чому можна навчитися</b>	Знати основні положення теорії штучних нейронних мереж, методологію побудови та використання глибоких нейронних мереж, методи застосування глибоких нейронних мереж для вирішення практичних задач. Уміння розробляти, тестувати та реалізовувати нейромережеві засоби з використанням мови програмування Python та бібліотек TensorFlow і Keras. Навички та досвід розв'язання типових задач розпізнавання образів
<b>Як можна користуватися набутими знаннями і уміннями</b>	Здатність розв'язувати професійні задачі аналізу та обробки даних за допомогою систем Deep Learning. Здатність розроблювати, експлуатувати та обслуговувати програмне забезпечення для інформаційних та інформаційно-управляючих систем Deep Learning різного призначення
<b>Інформаційне забезпечення дисципліни</b>	Монографії та онлайн-підручники, матеріали для виконання лабораторних робіт, тестові завдання
<b>Вид семестрового контролю</b>	Залік

<b>17. Інформаційні системи</b>	
<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	4, 7
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЕКТС: 54 аудиторні години, 66 годин самостійної роботи
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення дисципліни</b>	Базові знання об'єктно-орієнтованого програмування, основи баз даних, основи веб-розробки та тестування
<b>Що буде вивчатися</b>	Підходи до побудови інформаційних систем та програмних продуктів від бізнес проблеми до готової системи. Розподіл ролей та задач в команді при розробці проєкту. Методології роботи з функціональними та нефункціональними вимогами. Побудова архітектури системи на основі актуальних архітектурних стилів (наприклад, мікросервісів) та шаблонів. Підходи та тактики для забезпечення архітектурно-значущих вимог програмної системи.
<b>Чому це цікаво/треба вивчати</b>	Можливість відчувати себе командою у роботі над розробкою спільного проєкту за обраною темою групою до 3 студентів. Та спробувати себе в різних ролях: бізнес-аналітика, архітектора, розробника програмного забезпечення, UX/UI дизайнера чи менеджера проєкту. Напрацювання вмінь ефективно презентувати свою роботу та навичок командної комунікації. Вивчення прикладів, як правильно побудована архітектура системи та процесів розробки дозволяє суттєво зменшити витрати на створення, розвиток та експлуатацію програмного продукту, зробити продукт успішним і конкурентоспроможним.
<b>Чому можна навчитися</b>	Студенти можуть опанувати основні найбільш популярні підходи до побудови систем, навчитись обирати найбільш ефективні підходи для різних типів проєктів та обґрунтовувати свої рішення. Отримають досвід командної роботи і взаємодії, розуміння ролей всіх учасників розробки, та навчатись застосовувати гнучкі методології розробки. Матимуть змогу опанувати актуальні практики побудови мікросервісних систем та сучасні підходи до побудови стійких та надійних систем з високою доступністю.
<b>Як можна користуватися набутими знаннями і уміннями</b>	Набуті знання можуть бути застосовані як для побудови власного стартапу, так і при розробці корпоративних систем (для невеликих локальних бізнесів та міжнародних корпорацій). Дана дисципліна допоможе впевнено почуватися на співбесідах та ширше розуміти різні ролі та їх взаємодію в IT-індустрії.
<b>Інформаційне забезпечення дисципліни</b>	Монографії та онлайн-підручники, презентації лекцій, відеозаписи лекцій, матеріали для виконання лабораторних робіт, тестові завдання
<b>Вид семестрового контролю</b>	Залік

<b>18. Розподілені і хмарні обчислення</b>	
<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	4, 7
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
<b>Мова викладання</b>	українська
<b>Вимоги до початку вивчення дисципліни</b>	Знання з дисциплін «Архітектура обчислювальних систем», «Програмування», «Алгоритми і структури даних»
<b>Що буде вивчатися</b>	Основні поняття та концепції розподілених і хмарних обчислень, методів та підходів до розв'язання прикладних задач за допомогою розподілених і хмарних обчислень.
<b>Чому це цікаво/треба вивчати</b>	Завдяки об'єднанню ресурсів і непостійного характеру споживання з боку споживачів, розподілені і хмарні обчислення дозволяють економити на масштабах, використовуючи менші апаратні ресурси, ніж були потрібні б при виділених апаратних потужностях для кожного користувача.
<b>Чому можна навчитися</b>	<ul style="list-style-type: none"> <li>- мета побудови розподілених і хмарних систем;</li> <li>- поняття розподілених обчислень та розподіленої системи, хмарних обчислень та хмарних систем;</li> <li>- вимоги до розподілених і хмарних систем;</li> <li>- класифікація розподілених і хмарних систем та їх призначення;</li> <li>- переваги та недоліки розподілених і хмарних обчислень;</li> <li>- основні парадигми застосування розподілених і хмарних обчислень;</li> <li>- моделі розподілених і хмарних архітектур.</li> <li>- паралельні чисельні алгоритми для вирішення типових задач обчислювальної математики.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями</b>	<ul style="list-style-type: none"> <li>- розробляти моделі розподілених обчислень;</li> <li>- проводити аналіз комунікаційної трудомісткості розподілених алгоритмів;</li> <li>- аналізувати та обирати оптимальні рішення щодо залучення засобів розподілених і хмарних обчислень у напрямку їх застосування для проведення наукових досліджень, а також для створення єдиного обчислювального середовища організації, підприємства;</li> <li>- застосовувати базові знання стандартів в області інформаційних технологій під час розробки та впровадження розподілених обчислювальних систем на базі хмарних технологій та сервісів;</li> <li>- проектувати компоненти програмного забезпечення для роботи в якості сервісів у складі розподілених обчислювальних систем та комплексів й хмарних обчислень.</li> </ul>
<b>Інформаційне забезпечення дисципліни</b>	Слайди презентацій, класичні підручники, методичні посібники, онлайн підручник. Силабус
<b>Вид семестрового контролю</b>	Залік



<b>19. Навчання з підкріпленням у глибинному навчанні</b>	
<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	4, 7
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення дисципліни</b>	Базові знання з дисциплін: Математична логіка та теорія алгоритмів, Алгоритми і структури даних, Випадкові процеси, Аналіз даних, Вступ до баз даних та інформаційних систем, Основи машинного навчання, Методи оптимізації
<b>Що буде вивчатися</b>	Парадигми, моделі та методи глибинного навчання з підкріпленням
<b>Чому це цікаво/треба вивчати</b>	Навчання з підкріпленням (Reinforcement Learning) є актуальним напрямком в області Глибинного Навчання, оскільки дозволяє проводити навчання моделей у режимі постійної адаптації до складних середовищ великої розмірності і відсутності навчальної вибірки як такої.
<b>Чому можна навчитися</b>	Основи навчання з підкріпленням: методам динамічного програмування, Монте-Карло і SARSA (серед них, зокрема, Q-Learning). Методи глибинного навчання з підкріпленням: DQN, REINFORCE, Актор-Критик. Основи мультиагентного навчання з підкріпленням: методи незалежного навчання з підкріпленням
<b>Як можна користуватися набутими знаннями і уміннями</b>	Здатність розв'язувати професійні задачі збору, аналізу та обробки даних в умовах реального часу та відсутності навчальної вибірки. Здатність реалізовувати методи навчання з підкріплення для відповідної прикладної задачі і з урахуванням розмірності і типу характеристик середовища.
<b>Інформаційне забезпечення дисципліни</b>	Слайди презентацій, монографії, наукові статті, методичні посібники, онлайн підручники, силабус, матеріали для виконання лабораторних робіт, тестові завдання
<b>Вид семестрового контролю</b>	Залік

<b>20. Управління проектами</b>	
<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	4, 7
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення дисципліни</b>	Початковий досвід роботи у проектній команді (хоча б на рівні спільної підготовки та проведення якихось заходів, де кількість учасників перевищує одного)
<b>Що буде вивчатися</b>	Проектна документація: планування верхнього рівня. Загальні підходи до планування, структуризації і контролю проектів. Управління ресурсам проекту. Управління командою проекту. Ведення перемовин. Оцінка і контроль виконання проекту. Управління ризиками проекту.
<b>Чому це цікаво/треба вивчати</b>	Сучасні інформаційні системи створюються виключно проектними командами. Навіть якщо Ви ніколи не будете працювати керівником (менеджером) проекту, корисно розуміти його мотивацію та основні засоби управління проектними ресурсами. Головна мета дисципліни «Управління проектами» — дати студентам уявлення про методологію підготовки й реалізації, способи та засоби побудови проектів, залучення ресурсів для реалізації цих проектів і механізмів управління ними. Курс ґрунтується на міжнародних рекомендаціях з управління проектами та особистому досвіді лектора. Розглядається управління проектами у різних сферах (розробка програмного забезпечення, консалтинг, впровадження ІТ-рішень).
<b>Чому можна навчитися</b>	<ul style="list-style-type: none"> <li>• знання термінології управління проектами;</li> <li>• знання основних підходів до створення організаційних структур управління проектами;</li> <li>• знання методів мережного та календарного планування проектів;</li> <li>• уміння будувати структуру розподілу робіт у проекті;</li> <li>• уміння визначати логічну послідовність виконання робіт, складати та контролювати плани виконання робіт;</li> <li>• уміння визначати можливі ризики проекту та розробляти заходи з їх зниження;</li> <li>• досвід побудови проектного плану та визначення його критичного шляху.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями</b>	<ul style="list-style-type: none"> <li>• розуміння того, як започаткувати, управляти та завершувати проекти;</li> <li>• здатність адаптуватися під час управління проектом;</li> <li>• здатність оцінювати ризики проекту;</li> <li>• розуміння сучасних методів ведення перемовин.</li> </ul>
<b>Інформаційне забезпечення дисципліни</b>	Project Management Body of Knowledge (PMBOK), слайдові презентації лекцій, додаткова література та джерела з управління проектами та ведення перемовин
<b>Вид семестрового контролю</b>	Залік

<b>21. Застосування генеративного ШІ</b>	
<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	4, 7
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення дисципліни</b>	Знання з дисциплін «Архітектура обчислювальних систем», «Програмування», «Алгоритми і структури даних»
<b>Що буде вивчатися</b>	Основи генеративного штучного інтелекту, архітектури нейронних мереж, які використовуються для генерації контенту, методи навчання генеративних моделей, а також практичне застосування генеративного ШІ для створення тексту, зображень, відео та інших типів даних.
<b>Чому це цікаво/треба вивчати</b>	Генеративний штучний інтелект є однією з найдинамічніших технологій сучасності, що впливає на такі галузі, як мистецтво, дизайн, програмування, наука та бізнес-аналітика. Його вивчення дозволяє зрозуміти, як моделі ШІ можуть створювати тексти, зображення, музику, код програм та навіть креативні рішення для складних завдань. Завдяки цьому курсу студенти отримають знання про роботу передових нейромереж, таких як GPT, DALL·E та ін., навчатися розробляти та адаптувати генеративні моделі під реальні потреби. Вивчення цієї дисципліни надає конкурентні переваги на ринку праці, оскільки технології генеративного ШІ вже активно впроваджуються у великих технологічних компаніях, стартапах та дослідницьких центрах.
<b>Чому можна навчитися</b>	<ul style="list-style-type: none"> <li>- принципи роботи генеративних моделей, таких як автокодера (VAE), генеративно-змагальні мережі (GAN) та трансформери (GPT, BERT);</li> <li>- методи навчання генеративних нейронних мереж та їх оптимізацію;</li> <li>- особливості генерації тексту, зображень, аудіо та відео на основі ШІ;</li> <li>- застосування ШІ для оптимізації та вдосконалення коду програм;</li> <li>- застосування генеративного ШІ для автоматизації творчих процесів;</li> <li>- технічні та етичні аспекти використання генеративного ШІ.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями</b>	<ul style="list-style-type: none"> <li>- розробляти та навчати генеративні моделі для створення нових даних;</li> <li>- аналізувати та вдосконалювати моделі ШІ для підвищення їх ефективності;</li> <li>- застосовувати генеративний ШІ для створення контенту, автоматизації бізнес-завдань та вирішення наукових проблем;</li> <li>- інтегрувати генеративний ШІ у веб-додатки та мобільні сервіси;</li> <li>- розуміти ризики, пов'язані з використанням генеративних моделей, та застосовувати етичні стандарти.</li> </ul>
<b>Інформаційне забезпечення дисципліни</b>	Слайди презентацій, класичні підручники, методичні посібники, онлайн підручник. Силабус
<b>Вид семестрового контролю</b>	Залік

<b>22. Geometric Modelling</b>	
<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	4, 7
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
<b>Мова викладання</b>	Англійська
<b>Вимоги до початку вивчення дисципліни</b>	Familiarity with the topics covered in the following courses: Mathematical Analysis, Algebra and Geometry, Differential Equations, Numerical Methods, and Modelling of Complex Systems.
<b>Що буде вивчатися</b>	This course covers the construction of linear interpolation (for precisely defined points) and approximation (for points obtained with measurement errors) of planar and spatial splines. It also includes the development of interpolation and approximation surface splines in three-dimensional space. A key focus is on identifying optimal paths and trajectories—both in terms of energy efficiency and aesthetic appeal—subject to given constraints.
<b>Чому це цікаво/треба вивчати</b>	Geometry is fundamental to understanding natural phenomena, as well as to the design and analysis of machines, instruments, and structures. Mastery of geometric concepts is essential for calculating trajectories and designing roadways. The aesthetics of objects and contours are closely linked to geometric quantitative characteristics, which can be expressed as specific functionals. Students will actively engage in developing state-of-the-art methods for geometric modelling and optimisation.
<b>Чому можна навчитися</b>	The course will cover techniques for constructing aesthetically pleasing curves and surfaces that pass through specified points. Students will learn to smooth measurement errors in geometric structures or trajectories, reconstructing accurate geometry while accounting for predefined features. The course also includes formulating and solving mathematical functionals to identify the most efficient geometric solutions and establishing accuracy requirements for measuring instruments.
<b>Як можна користуватися набутими знаннями і уміннями</b>	The concepts taught in this course are applicable in: Reverse engineering, particularly in processing scanned data of manufactured objects. Computer graphics, including static and motion modelling. Image recognition and smoothing, improving the quality of visual representations. Industrial and product design, creating aesthetically refined solutions. Trajectory optimisation, determining energy-efficient paths for moving objects in both planar and three-dimensional space. Quality assessment, ensuring the precision and accuracy of constructed designs.
<b>Інформаційне забезпечення дисципліни</b>	1. Syllabus 2. Rogers, D. F. (2000). An Introduction to NURBS: With Historical Perspective. Elsevier. 3. Höllig, K., & Hörner, J. (2013). Approximation and Modelling with B-Splines. Society for Industrial and Applied Mathematics.
<b>Вид семестрового контролю</b>	Залік

## 23. Системний аналіз

<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	4, 8
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЄКТС: 72 аудиторні години, 48 годин самостійної роботи
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення дисципліни</b>	Опанування курсів теорії ймовірностей, математичної статистики, аналізу даних, методів оптимізації, дослідження операцій, математичного моделювання
<b>Що буде вивчатися</b>	Принципи, процедури, методи та методології системного аналізу. Дослідження складних та надскладних систем. Системно-методологічні аспекти моделювання.
<b>Чому це цікаво/треба вивчати</b>	Одна з найактуальніших проблем нової епохи – проблема ефективного управління складними системами. Завдання, що постають в процесі створення та функціонування складних та надскладних систем, вимагають володіння системним мисленням, спеціально розробленими методами та методологіями.
<b>Чому можна навчитися</b>	Знання предметної області та основних понять і методів системного аналізу. Уміння розкривати невизначеності; знаходити раціональний компроміс; проводити інформаційний аналіз системних задач і структурно-функціональний аналіз складних ієрархічних систем; передбачати та прогнозувати поведінку систем. Досвід аналізу систем, проектування та розробки додатків для розв'язання системних задач.
<b>Як можна користуватися набутими знаннями і уміннями</b>	Проектувати та досліджувати складні системи. Створювати моделі складних систем. Діяти в умовах нестачі інформації, суперечливості інформації, надлишку інформації. Приймати складні рішення щодо функціонування систем.
<b>Інформаційне забезпечення дисципліни</b>	Онлайн підручники та курси лекцій, слайди презентацій, підручники, методичні посібники, силабус
<b>Вид семестрового контролю</b>	Залік

<b>24. Теорія оптимального керування</b>	
<b>Кафедра, яка забезпечує викладання</b>	прикладної математики
<b>Рівень вищої освіти</b>	Перший (бакалаврський)
<b>Курс, семестр</b>	4, 8
<b>Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи</b>	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення дисципліни</b>	Базові знання з дисциплін: Математичний аналіз, лінійна алгебра, диференціальні рівняння, теорія функцій комплексної змінної, методи оптимізації, числові методи, алгоритми і системи комп'ютерної математики, програмування.
<b>Що буде вивчатися</b>	Теорія керування – це наука про принципи й методи спрямування функцій різних систем, процесів та об'єктів. Її сутність полягає в побудові математичної моделі об'єкта керування на основі його системного аналізу та в синтезі алгоритму для досягнення бажаних характеристик функціонування або цілей керування.
<b>Чому це цікаво/треба вивчати</b>	У даний час ця теорія слугує єдиною науковою основою розв'язування задач керування динамічними системами різної природи (фізичної, хімічної, біологічної та соціальної тощо) на базі розвинених математичних методів їх дослідження, зокрема їх аналізу та синтезу. Сучасна теорія керування є послідовницею кібернетики, науки що вивчає управління різноманітними системами живої та неживої природи на основі принципу зворотного зв'язку. Теорія керування має глибокі зв'язки з теорією нейронних мереж.
<b>Чому можна навчитися</b>	Будувати моделі динамічних систем у вигляді диференціальних або ітеративних рівнянь та систем рівнянь, ідентифікувати ці моделі на даних, будувати критерії їх оптимального функціонування, знаходити стратегії оптимального керування, чисельно розв'язувати задачі оптимального керування.
<b>Як можна користуватися набутими знаннями і уміннями</b>	Здатність розв'язувати професійні задачі проектування керованих технічних систем, в тому числі автономних керованих систем (роботів, дронів) в програмних середовищах математичного моделювання (Matlab, Octava, Scilab і т.і.).
<b>Інформаційне забезпечення дисципліни</b>	Конспекти лекцій, підручники та монографії. Силабус Новицький І.В. Сучасна теорія керування: навч. посіб. / І.В. Новицький, С.А. Ус, м-во освіти і науки України, Нац. гірн. ун-т. – Дніпро : НГУ, 2017. – 263 с. Пічкур В. В. Лекції з теорії керування. Київ: Київський національний університет імені Тараса Шевченка, Факультет комп'ютерних наук та кібернетики, 2017. – 232 с. Paluszek M., Thomas S. MATLAB Machine Learning Recipes: A Problem-Solution Approach. 2nd ed. APRESS, 2019.
<b>Вид семестрового контролю</b>	Залік

<b>25. Методи теорії надійності та ризику</b>	
Кафедра, яка забезпечує викладання	прикладної математики
Рівень вищої освіти	Перший (бакалаврський)
Курс, семестр	4, 8
Обсяг дисципліни та розподіл годин аудиторної та самостійної роботи	4 кредити ЄКТС: 54 аудиторні години, 66 годин самостійної роботи
Мова викладання	Українська
Вимоги до початку вивчення дисципліни	Знання математичного аналізу, теорії ймовірностей, математичної статистики, дослідження операцій, теорії та методів оптимізації.
Що буде вивчатися	<p>Невизначеність, непередбачуваність та ризику є невід'ємними атрибутами людської діяльності та навколишнього світу. В дисципліні <b>Методи теорії надійності та ризику</b> будуть вивчатися:</p> <ul style="list-style-type: none"> <li>● Моделі і методи прийняття оптимальних рішень та керування в умовах невизначеності та ризику;</li> <li>● математичні моделі оцінки та мінімізації страхових та фінансових ризиків;</li> <li>● методи розрахунку та оптимізації надійності роботи складних багатокомпонентних систем,</li> <li>● моделі і методи захисту критичної інфраструктури від терористичних атак.</li> </ul>
Чому це цікаво/треба вивчати	<p>1. Сучасна теорія (мір) ризику є основою страхової та фінансової математики.</p> <p>2. Проблеми захисту критичної інфраструктури та підвищення її стійкості є надзвичайно важливими. Особливо це актуально в нинішній час, коли система міжнародної безпеки зруйнована, а ризику військових, терористичних, інформаційних та інших загроз різко зросли. До того ж, глобальні кліматичні процеси та складність технічних систем істотно підвищують загрози аномальних природних явищ і потенційних техногенних катастроф.</p>
Чому можна навчитися	<p>Розробляти та застосовувати математичні моделі та кількісні методи системного аналізу, дослідження операцій, теорії ігор, теорії надійності та ризику для:</p> <ul style="list-style-type: none"> <li>● підтримки прийняття фінансових рішень,</li> <li>● підвищення захисту та стійкості об'єктів і систем критичної інфраструктури до інтенсивних зловмисних атак.</li> </ul>
Як можна користуватися набутими знаннями і уміннями	<p>Здатність розв'язувати професійні задачі:</p> <ul style="list-style-type: none"> <li>● обґрунтування фінансових рішень,</li> <li>● моделювання та планування захисту систем критичної інфраструктури,</li> <li>● проектування керованих технічних систем, що функціонують в умовах невизначеності та ризику в програмних середовищах математичного моделювання (Matlab, Octava, Scilab і т.і.).</li> </ul>
Інформаційне забезпечення дисципліни	<p>Конспекти лекцій, підручники та монографії.</p> <p><u>Силабус.</u></p> <p><u>Paluszek M., Thomas S. MATLAB Machine Learning Recipes: A Problem-Solution Approach. 2nd ed. APRESS, 2019.</u></p>
Вид семестрового контролю	Залік