



**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»**

ЗАТВЕРДЖЕНО:

Методичною радою  
КПІ ім. Ігоря Сікорського  
(протокол № 5 від «06» 03. 2025 р.)

**Міжфакультетський КАТАЛОГ  
вибіркових навчальних дисциплін  
третього (PhD) 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 «27» січня 2025 р.)

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

**Київ 2025**

## Зміст

Інструкція користувачам каталогу	3
Ф-каталог	4
Анотації вибіркових дисциплін	
1. Технології віртуалізації та хмарних обчислень	5
2. Гетерогенні розподілені бази даних	7
3. Мультиагентно-орієнтоване програмування	8
4. Прикладні методи аналізу даних	10
5. Проектування програмно-конфігурованих мереж	12
6. Проектування програмного забезпечення технології цифрових двійників	13
7. Методи побудови програмних засобів для матричних обчислень	15
8. Обчислювальні системи високої готовності	16
9. Відмовостійкі багатопроцесорні системи	18
10. Децентралізовані застосунки	20
11. Математичні методи інформаційних систем	21

## Інструкція користувачам каталогу

1. Кількість і обсяг (у кредитах ЄКТС) навчальних дисциплін, які може обрати аспірант (вибіркових дисциплін) визначається навчальним планом, а саме для 3 семестру – 5 кредитів (одна дисципліна), для 4 семестру – 5 кредитів (одна дисципліна).
2. Процедура вибору аспірантами навчальних дисциплін з Ф-каталогу реалізується через спеціалізовану інформаційну систему Університету [my.kpi.ua](http://my.kpi.ua) та включає такі етапи:
  - 2.1 Реєстрація аспіранта в [my.kpi.ua](http://my.kpi.ua)
  - 2.2 Здійснення аспірантом вибору дисциплін
  - 2.3 Підтвердження аспіранту його вибору навчальних дисциплін із Ф-Каталогу.
  - 2.4 Опрацювання результатів вибору дисциплін (фіксація результатів вибору) та передача даних для корекції індивідуальних навчальних планів аспірантів.
3. У разі неможливості формування академічної групи для вивчення певної вибіркової дисципліни, аспірантам надається можливість здійснити повторний вибір, приєднавшись до вже сформованих навчальних груп.
4. Аспірант, який знехтував своїм правом вибору, буде записаний на вивчення тих дисциплін, які завідувач випускової кафедри вважатиме потрібними для оптимізації навчальних груп і потоків.
5. Якщо здобувач із поважної причини не зміг обрати дисципліни вчасно, або виявив помилку щодо свого волевиявлення, він звертається до деканату із заявою для запису на вивчення обраних ним дисциплін, надавши документів, які засвідчують поважність причин. Заява на зміну вибіркової дисципліни у сформованому індивідуальному навчальному плані має подаватися не пізніше ніж за місяць до початку семестру, в якому викладається ця дисципліна.
6. Не допускається зміна обраних дисциплін після початку навчального семестру, в якому вони викладаються.
7. Обрані аспірантом навчальні дисципліни зазначаються у його індивідуальному навчальному плані.

## Ф-Каталог

### *Дисципліни для 2 курсу (вибір першокурсниками)*

Потрібно обрати 10 кредитів за рік:

3-й семестр - 5 кредитів;

4-й семестр - 5 кредитів.

№	Назва навчальної дисципліни	Семестр	Кіл-ть кредитів	Семестрова атестація
1.	Технології віртуалізації та хмарних обчислень (ІПІ)	3	5	Залік
2.	Гетерогенні розподілені бази даних (ІСТ)	3	5	Залік
3.	Мультиагентно-орієнтоване програмування (ІПІ)	3	5	Залік
4.	Прикладні методи аналізу даних (ОТ)	3	5	Залік
5.	Проектування програмно-конфігурованих мереж (ОТ)	3	5	Залік
6.	Проектування програмного забезпечення технології цифрових двійників (ПЗКС)	4	5	Залік
7.	Методи побудови програмних засобів для матричних обчислень (ПЗКС)	4	5	Залік
8.	Обчислювальні системи високої готовності (СПСКС)	4	5	Залік
9.	Відмовостійкі багатопроцесорні системи (СПСКС)	4	5	Залік
10.	Децентралізовані застосунки (ННІАТЕ)	4	5	Залік
11.	Математичні методи інформаційних систем (ІПІ)	4	5	Залік

## Анотації вибіркових дисциплін

Дисципліна	Технології віртуалізації та хмарних обчислень
<b>Кафедра</b>	Інформатики та програмної інженерії ФІОТ
<b>Рівень ВО</b>	Третій (PhD)
<b>Курс, семестр</b>	2 курс, 3 семестр 39 год. аудиторної роботи, 111 год. СРС
<b>Обсяг дисципліни</b>	5 кредитів ЄКТС
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Успішному вивченню дисципліни «Технології віртуалізації та хмарних обчислень» передуює вивчення дисципліни «Формальні методи програмної інженерії» навчального плану підготовки докторів філософії за спеціальністю 121 Інженерія програмного забезпечення.
<b>Що буде вивчатися</b>	<ul style="list-style-type: none"> <li>- освоєння теоретичних понять і практичних навичок у галузі розроблення і експлуатації інфраструктури ЦОД;</li> <li>- ознайомлення з сучасними методами і технологіями управління віртуалізованими і хмарними системами;</li> <li>- набуття навичок побудови ІТ-інфраструктури для ЦОД та хмарних середовищ.</li> </ul>
<b>Чому це цікаво/треба вивчати</b>	Мета вивчення дисципліни - набуття ключових фахових компетентностей, теоретичних знань і практичних навичок з технології віртуалізації та хмарних обчислень. Предметом вивчення дисципліни є методи і технології управління віртуалізованими і хмарними системами, засоби побудови ІТ-інфраструктури центрів оброблення даних (ЦОД) та хмарних середовищ.
<b>Чому можна навчитися (результати навчання)</b>	Отримані в результаті засвоєння дисципліни «Технології віртуалізації та хмарних обчислень» теоретичні знання та практичні уміння можуть бути корисні для проведення наукових досліджень за темою дисертації.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<p>Після засвоєння дисципліни «Технології віртуалізації та хмарних обчислень» результатами навчання є:</p> <p>знання:</p> <ul style="list-style-type: none"> <li>- сучасних теоретичних, методичних і алгоритмічних основ розробки віртуалізованих і хмарних систем для їх використання під час розв'язання прикладних і наукових задач;</li> <li>- визначальні характеристики, сервісні моделі та моделі розгортання хмарних обчислень;</li> <li>- з моделювання стану та поведінки складних об'єктів комп'ютерної галузі в процесі тестування відповідних програмних і технічних систем при забезпеченні якості і надійності розроблених інформаційних систем і технологій;</li> <li>- математичних моделей і методів аналізу та синтезу сучасних систем управління та їх фундаментальних властивостей при проектуванні ІТ-інфраструктур для дослідження характеристик і поведінки складних об'єктів і систем;</li> </ul>

	<ul style="list-style-type: none"> <li>- архітектури та стандартів компонентних моделей, комунікаційних засобів і розподілених обчислень, уміння розв'язувати проблеми масштабованості, підтримки віддалених компонентів і взаємодії різних програмних платформ у віртуалізованих і хмарних інформаційних системах.</li> <li>- уміння: <ul style="list-style-type: none"> <li>- використовувати методи та методики проведення наукових та прикладних досліджень;</li> <li>- розв'язувати прикладні задачі в галузі проектування, налаштування та експлуатації віртуалізованих і хмарних систем;</li> <li>- використовувати методи проведення досліджень та вміти аналізувати складність технічних систем, розуміти складність задач оптимізації цих систем та їх елементів, та вдосконалювати методики їх проведення;</li> <li>- застосовувати різні парадигми програмування: структурне, об'єктно-орієнтоване, функціональне, логічне, з відповідними моделями, методами та алгоритмами обчислень, структурами даних і механізмами управління під час створення та удосконалення віртуалізованих і хмарних систем;</li> <li>- застосовувати технічні і програмні комунікаційні засоби та засоби з'єднання ресурсів в гібридних хмарних системах;</li> <li>- проектувати IT-інфраструктуру ЦОД для підтримки хмарних обчислень з використанням сучасних апаратно-програмних засобів із забезпеченням надійності, доступності та безпеки;</li> <li>- освоювати інноваційні інформаційні технології у галузі хмарних обчислень, оцінювати їх та використовувати з метою розробки архітектури та проектування хмарних послуг.</li> </ul> </li> <li>- досвід: <ul style="list-style-type: none"> <li>- проектування віртуалізованих і хмарних систем, розроблення сучасних інформаційних технологій та програмних систем для реалізації хмарних обчислень;</li> <li>- здатність використовувати професійні знання й уміння в галузі практичного використання хмарних технологій.</li> </ul> </li> </ul>
<b>Інформаційне забезпечення</b>	Силабус Електронний конспект лекцій
<b>Семестровий контроль</b>	Залік

<b>Дисципліна</b>	<b>Гетерогенні розподілені бази даних</b>
<b>Кафедра</b>	Інформаційних систем та технологій ФІОТ
<b>Рівень ВО</b>	Третій (PhD)
<b>Курс, семестр</b>	2 курс, 3 семестр 39 год. аудиторної роботи, 111 год. СРС
<b>Обсяг дисципліни</b>	5 кредитів ЄКТС
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Володіння знаннями із дисциплін «Розподілені системи обробки даних», «Паралельні та розподілені обчислення», «Хмарні та Grid-технології», «Бази даних»
<b>Що буде вивчатися</b>	<ul style="list-style-type: none"> <li>– Концепції та основи побудови розподілених баз даних;</li> <li>– розробка архітектури розподілених баз даних;</li> <li>– побудова гетерогенних розподілених систем;</li> <li>– технології управління гетерогенними системами;</li> <li>– технології обробки великих масивів даних в розподілених базах даних;</li> <li>– управління гетерогенними розподіленими базами даних.</li> </ul>
<b>Чому це цікаво/треба вивчати</b>	Для задоволення потреб обробки інформації, що зберігається в різномірних базах даних, які об'єднуються в одну розподілену систему в науковій, метеорологічній, астрофізичній, економічній, банківській, фізико-технічній сферах і набуття досвіду та уміння використання та створення систем розподіленої обробки даних з баз даних; для використання та розроблення програмного забезпечення з використанням сучасних засобів зберігання та обробки даних; для створення та застосування нових технологій доступу до даних через бездротовий зв'язок, зокрема технології 5G.
<b>Чому можна навчитися (результати навчання)</b>	<ul style="list-style-type: none"> <li>– Використовувати та організовувати розподілену обробку даних;</li> <li>– аналізувати вимоги до сучасних баз даних;</li> <li>– застосовувати та створювати методи оцінки продуктивності розподілених систем;</li> <li>– розв'язувати задачі проектування та експлуатації розподілених баз даних;</li> <li>– розробляти нові архітектури систем з високою швидкістю обробки даних;</li> <li>– адмініструвати розподілені бази даних;</li> <li>– застосовувати теоретичні знання для розв'язання практичних задач обробки даних.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Набути досвід практичної роботи в розподілених системах обробки даних, освоїти існуюче програмне забезпечення різномірних баз даних та будувати власні програмні продукти з використання гетерогенних розподілених баз даних, розпаралелювати доступ та обробку даних.

<b>Інформаційне забезпечення</b>	Силабус Електронний конспект лекцій
<b>Семестровий контроль</b>	Залік



<b>Дисципліна</b>	<b>Мультиагентно-орієнтоване програмування</b>
<b>Кафедра</b>	Інформатики та програмної інженерії ФІОТ
<b>Рівень ВО</b>	Третій (PhD)
<b>Курс, семестр</b>	2 курс, 3 семестр
<b>Обсяг дисципліни</b>	5 кредитів ЄКТС 39 год. аудиторної роботи, 111 год. СРС
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Дисципліна "Мультиагентно-орієнтоване програмування" базується на підготовці магістерського рівня зі спеціальності 121 «Інженерія програмного забезпечення».
<b>Що буде вивчатися</b>	<ul style="list-style-type: none"> <li>- Мультиагентні системи і введення до мультиагентно-орієнтованого програмування;</li> <li>- Агентні моделі і програмування їх;</li> <li>- Моделі середовища і їх програмування в сучасному інструментарії;</li> <li>- Організаційні моделі і покрокове програмування мультиагентних організацій.</li> </ul>
<b>Чому це цікаво/треба вивчати</b>	<p>Сучасні програмні додатки мають справу із зростаючим рівнем автономності взаємопов'язаних програмних систем, і перш за все з інтеграцією незліченних систем, які не відомі заздалегідь. Сучасні тенденції, такі як розумні міста, інтелектуальні транспортні системи та промисловість, сприяння розвитку Інтернет-речей, наприклад, вказують на ще більш складні сценарії, в яких адаптивні та відкриті команди інтелектуальних автономних суб'єктів програмного забезпечення та робітників будуть взаємодіяти з люди та побутові предмети, всі взаємопов'язані. Багатоагентні системи (БАС) можуть бути використані як відповідна парадигма для моделювання та проектування таких систем. Багатоагентна система - це організований ансамбль автономних цілеспрямованих сутностей, які називаються агентами, що взаємодіють між собою та взаємодіють в середовищі. На індивідуальному рівні кожен агент може мати власні цілі та завдання, які слід переслідувати самостійно, вирішуючи, які дії робити. Як ансамбль, агентам, як правило, потрібно координуватися та співпрацювати для досягнення глобальної мети БАС в цілому, як організації. У цій дисципліні йдеться про програмування багатоагентних систем із використанням інтегрованого підходу, який зараз називають мультиагентним орієнтованим програмуванням (МАОП).</p> <p>У літературі багато відповідних методів роботи з мультиагентними системами виникло в різному контексті - основними прикладами є штучний інтелект (ШІ), розподілений ШІ, програмна інженерія (СЕ), моделювання - і деякі з них призвели до конкретних моделей програмування для боротьби зі зростаючим рівнем автономності та складності взаємодій у сучасних системах. У цьому напрямку МАОП забезпечує структурований підхід, заснований на трьох взаємопов'язаних наборах концепцій та абстракцій програмування (надалі називаються вимірами), корисних для проектування таких складних систем: розмір агента, який використовується для програмування окремих (взаємодіючих) автономних сутностей; вимір навколишнього</p>

	<p>середовища, який використовується для програмування спільних ресурсів та засобів, що використовуються агентами для роботи, взаємодії та підключення до реального світу; та організаційний вимір, що використовується для структурування та регулювання складних взаємозв'язків, що відбуваються між автономними агентами у спільному середовищі.</p> <p>У цьому курсі з багатоагентного програмування основна увага приділяється практичному програмуванню багатоагентних систем. Він стосується одного з основних напрямків використання мультиагентної парадигми для розробки складних систем: мультиагентно-орієнтованого програмування (МАОР), що використовується для інтеграції інтелектуальних систем та розвитку соціально-технічних систем.</p> <p>Окрім вивчення основ та найсучасніших робіт, зроблених за цією темою в домені Multi-Agent, ви дізнаєтесь, як застосовувати підхід МАОР для розробки децентралізованих та відкритих систем AI.</p>
<b>Чому можна навчитися (результати навчання)</b>	Навчання використування платформи JaCaMo/Jason Agent Programming Language для практичних вправ, які представлені в цьому курсі. Ця платформа програмування є результатом інтеграції мови програмування агента Jason Agent Programming Language, платформи програмування середовища CARTaGo Environment Programming platform та платформи програмування MOISE Organizationprogrammingplatform.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>– Багатоагентний підхід, орієнтований на програмування, полягає у складі моделей агентів, моделей середовища та моделей організації, що взаємодіють між собою. Кожна з цих моделей побудована шляхом вибору концепцій відповідно до виміру Агент, Середовище чи Організація.</li> <li>– Використання МАОР шляхом надання привілеїв кожному з цих вимірів може відповідно призвести до Агентно-орієнтованого програмування (AOP), Орієнтованого на середовище програмування (EOP) та Організаційно-орієнтованого програмування (Organization Oriented Programming).</li> </ul>
<b>Інформаційне забезпечення</b>	<p>Силабус</p> <p>Електронний конспект лекцій</p> <p>Веб-сторінка дисципліни:</p> <p><a href="http://baklaniv.at.ua/index/multiagentno_orientovane_programuvannja/0-38">http://baklaniv.at.ua/index/multiagentno_orientovane_programuvannja/0-38</a></p>
<b>Семестровий контроль</b>	Залік

<b>Дисципліна</b>	<b>Прикладні методи аналізу даних</b>
<b>Кафедра</b>	Обчислювальної техніки ФІОТ
<b>Рівень ВО</b>	Третій (PhD)
<b>Курс, семестр</b>	2 курс, 3 семестр 39 год. аудиторної роботи, 111 год. СРС
<b>Обсяг дисципліни</b>	5 кредитів ЄКТС
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з вищої математики (математичний аналіз, лінійна алгебра та аналітична геометрія), теорія імовірностей та математична статистика, навички програмування алгоритмічною мовою Python
<b>Що буде вивчатися</b>	Основи сучасних технологій обробки даних (Data Science) та алгоритми, які лежать в основі пошуку прихованих функціональних залежностей в наборах даних, отриманих шляхом накопичення та попередньої обробки. Для ефективної реалізації розглянутих технологій обробки даних будуть вивчатися прикладні бібліотеки та програмні системи, зокрема, відповідні модулі у складі мови програмування Python, дистрибутив Anaconda з набором вбудованих інструментів для обробки даних, інтерактивна командна оболонка Jupyter Notebook. Окрема увага буде приділена вивченню бібліотеки Pandas як базового інструментарію для добування даних та їх аналізу.
<b>Чому це цікаво/треба вивчати</b>	Велика кількість організацій, які за характером своєї діяльності накопичили значні обсяги даних про клієнтів, все більше проявляють інтерес до використання цих даних з бізнесовими та дослідницькими цілями. В зв'язку з цим виникає великий спектр задач з визначення корисної інформації, на основі якої можна отримати знання, які не можуть бути одержані у інший спосіб.
<b>Чому можна навчитися (результати навчання)</b>	<ul style="list-style-type: none"> <li>– Методам формування наборів даних, які є зручними для вирішення конкретних прикладних задач;</li> <li>– використанню популярних програмних засобів для добування даних;</li> <li>– завантажувати та фрагментувати великі набори даних для оптимальної їх обробки та аналізу;</li> <li>– профілювати дані для представлення їх різними способами візуалізації;</li> <li>– відображати результати дослідження даних найбільш зручним чином.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>– Для перетворення «сирих» наборів початкових даних в такі оптимізовані за розміром набори даних, що є зручними для подальшого аналізу;</li> <li>– для виконання обробки даних з застосуванням розглянутих у рамках даного курсу засобів попередньої обробки;</li> <li>– для визначення невідомих функціональних залежностей та їх композицій у попередньо підготовлених наборах даних;</li> <li>– для тестування та перевірки моделей, побудованих на базі отриманих композицій функціональних залежностей;</li> <li>– для зручного та наглядного відображення отриманих результатів, які забезпечують оптимальність подальшого прийняття рішень.</li> </ul>

<b>Інформаційне забезпечення</b>	Силабус Електронний конспект лекцій
<b>Семестровий контроль</b>	Залік

<b>Дисципліна</b>	<b>Проектування програмно-конфігурованих мереж</b>
<b>Кафедра</b>	Обчислювальної техніки ФІОТ
<b>Рівень ВО</b>	Третій (PhD)
<b>Курс, семестр</b>	2 курс, 3 семестр 39 год.аудиторної роботи, 111 год. СРС
<b>Обсяг дисципліни</b>	5 кредитів ЄКТС
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Базові знання з вищої математики, програмування, дискретної математики, комп'ютерних мереж.
<b>Що буде вивчатися</b>	Основні теоретико-методичні та практичні аспекти побудови програмно-конфігурованих мереж (SDN), контролер SDN, мобільна мережа SDN, протокол OpenFlow, інтерфейс прикладних програм, віртуалізація мережевих функцій (API), система моделювання SDN мереж Mininet.
<b>Чому це цікаво/треба вивчати</b>	Програмно-конфігуровані мережі ефективні для побудови віртуальних мереж, інфраструктурних хмарних сервісів, великих центрів обробки даних. Дозволяють за запитом від споживачів послуг необхідно автоматично і в найкоротші терміни створювати віртуальні вузли і виділяти віртуальні мережеві ресурси для них.
<b>Чому можна навчитися (результати навчання)</b>	Засвоїти принципи розгортання, налаштування та використання основних елементів програмно-конфігурованих мереж за допомогою системи моделювання Mininet.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Здатність підібрати структуру, планувати і реалізовувати SDN мережі, керувати мережними ресурсами, підібрати комплекс необхідних апаратно-програмних засобів для комп'ютерної мережі, розширювати і модернізувати мережі.
<b>Інформаційне забезпечення</b>	Силабус Електронний конспект лекцій
<b>Семестровий контроль</b>	Залік

<b>Дисципліна</b>	<b>Проектування програмного забезпечення технології цифрових двійників</b>
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем ФПМ
<b>Рівень ВО</b>	Третій (PhD)
<b>Курс, семестр</b>	2 курс, 4 семестр 36 год. аудиторної роботи, 114 год. СРС
<b>Обсяг дисципліни</b>	5 кредитів ЄКТС
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Успішному вивченню дисципліни «Проектування програмного забезпечення технології цифрових двійників» передуює вивчення дисципліни «Формальні методи програмної інженерії» навчального плану підготовки докторів філософії за спеціальністю 121 Інженерія програмного забезпечення.
<b>Що буде вивчатися</b>	<ul style="list-style-type: none"> <li>– Технологія цифрових двійників;</li> <li>– теоретичні основи подання та оброблення даних цифрових двійників;</li> <li>– програмне забезпечення технології цифрових двійників.</li> </ul>
<b>Чому це цікаво/треба вивчати</b>	Цифровий двійник являє собою модель фізичного об'єкта або процесу (фізичного двійника), яка якнайповніше відображає його характеристики у динаміці. Технологія цифрових двійників передбачає подання, оброблення, маніпулювання усіма даними, що характеризують фізичного двійника, та отримання певних компонент цих даних по мірі виникнення потреби у них при вирішенні задач аналізу та прогнозування стану фізичного двійника, оптимізації процедур керування фізичним двійником тощо. Технологія цифрових двійників визнана однією з перспективних технологій для застосування у різних галузях людської діяльності.
<b>Чому можна навчитися (результати навчання)</b>	<p>Після засвоєння дисципліни «Проектування програмного забезпечення технології цифрових двійників» результатами навчання є знання:</p> <ul style="list-style-type: none"> <li>– принципів створення цифрових двійників;</li> <li>– видів цифрових двійників досліджуваних об'єктів;</li> <li>– теоретичних засад технології цифрових двійників;</li> <li>– вимог до програмного забезпечення технології цифрових двійників;</li> <li>– принципів проектування програмного забезпечення технології цифрових двійників;</li> </ul> <p>та уміння:</p> <ul style="list-style-type: none"> <li>– аналізувати особливості об'єкта дослідження та формувати вимоги до програмного забезпечення для створення цифрового двійника певного об'єкта дослідження;</li> <li>– проектувати архітектуру програмного забезпечення технології цифрових двійників;</li> <li>– проектувати компоненти програмних продуктів на основі технології цифрових двійників.</li> </ul>

<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>- Розв'язання складних задач професійної діяльності, пов'язаної із розробленням програмних систем із застосуванням технології цифрових двійників;</li> <li>- проектування архітектури програмного забезпечення для створення цифрового двійника певного об'єкта дослідження.</li> </ul>
<b>Інформаційне забезпечення</b>	Силабус Електронний конспект лекцій
<b>Семестровий контроль</b>	Залік

<b>Дисципліна</b>	<b>Методи побудови програмних засобів для матричних обчислень</b>
<b>Кафедра</b>	Програмного забезпечення комп'ютерних систем ФПМ
<b>Рівень ВО</b>	Третій (PhD)
<b>Курс, семестр</b>	2 курс, 4 семестр
<b>Обсяг дисципліни</b>	5 кредитів ЄКТС 36 год. аудиторної роботи, 114 год. СРС
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Дисципліна «Методи побудови програмних засобів для матричних обчислень» базується на підготовці бакалаврського та магістерського рівня зі спеціальності 121 «Інженерія програмного забезпечення».
<b>Що буде вивчатися</b>	<ul style="list-style-type: none"> <li>– Методи розв'язання часткової проблеми власних значень;</li> <li>– методи розв'язання повної проблеми власних значень;</li> <li>– метод виконання сингулярного розкладення прямокутної матриці;</li> <li>– методи апроксимації таблично-заданих функцій;</li> </ul> методи програмного розв'язання систем нелінійних рівнянь.
<b>Чому це цікаво/треба вивчати</b>	Матричні обчислення є важливою складовою у широкому колі досліджень, тому розроблення програмних засобів для матричних обчислень може бути корисним багатьом аспірантам.
<b>Чому можна навчитися (результати навчання)</b>	<ul style="list-style-type: none"> <li>– Аналізувати програмні методи виконання матричних обчислень;</li> <li>– аналізувати та оцінювати похибки проведених обчислень над матрицями великої розмірності;</li> <li>– виконувати вибір необхідних математичних та програмних функцій для розв'язання поставленої задачі;</li> <li>– розробляти та ефективно застосувати програмні додатки для чисельного розв'язання матричних задач.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>– Проектування систем комп'ютерної математики для виконання матричних обчислень;</li> <li>– використання сучасних математичних пакетів.</li> </ul>
<b>Інформаційне забезпечення</b>	Силабус Електронний конспект лекцій
<b>Семестровий контроль</b>	Залік



<b>Дисципліна</b>	<b>Обчислювальні системи високої готовності</b>
<b>Кафедра</b>	Системного програмування і спеціалізованих комп'ютерних систем ФПМ
<b>Рівень ВО</b>	Третій (PhD)
<b>Курс, семестр</b>	2 курс, 4 семестр
<b>Обсяг</b>	5 кредитів ЄКТС 36 год. аудиторної роботи, 114 год. СРС
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Знання та вміння в межах програми магістра Комп'ютерної інженерії; знання та розуміння принципів, концепцій та технологій побудови систем обробки даних;
<b>Що буде вивчатися</b>	методи, засоби та процеси аналізу, моделювання, проектування та конструювання розподілених систем обробки даних (PCOD)
<b>Чому це цікаво/треба вивчати</b>	<p>Питання про використання розподілених систем обробки даних стало актуальним з появою потужних обчислювальних систем з розподіленими ресурсами в межах одного комп'ютера (багаторядні системи), локальних корпоративних і зовнішніх (регіональних і глобальних) мереж, кластерів, технологій пошуку і багатовимірного аналізу даних, розвитком Web-технологій .</p> <p>Загальна інформатизація управлінських та виробничих організацій міняє сам підхід до використання PCOD. З засобу накопичення даних про їх діяльність PCOD перетворюється на інструмент підвищення ефективності управління розподіленими організаціями та виробництвами. Вона прискорює процес аналітичної обробки даних та надає різносторонню інформацію для прийняття рішень, а життєздатність організацій збільшується за рахунок підвищення ефективності рішень, що приймаються на основі інформації, наданої PCOD, та можливості оперативного спілкування з дієвими особами оточуючого середовища;</p> <p>Ні одна географічно розсереджена організація сьогодні не може існувати без ефективного PCOD.</p> <p>Сучасна PCOD має складну багаторівневу структуру. Її основу складають інтегровані та розподілені реляційні та об'єктні бази даних, інтегровані та розподілені об'єктно-орієнтовані програмні компоненти та засоби віддаленого доступу до інформації.</p> <p>Вивчення цієї дисципліни надає комплексне уявлення про різнобічні технології та засоби, що використовуються на кожному рівні моделювання, проектування та реалізації PCOD і дозволяє достойно представляти себе на найбільшому сегменті IT-ринку праці;</p> <p>Спеціалісти в галузі аналізу, моделювання, проектування та реалізації PCOD користуються підвищеним попитом на ринку праці, а самі PCOD постійно розвиваються та оновлюються і вимагають від розробника володіння найбільш сучасними IT технологіями.</p>
<b>Чому можна навчитися (результати навчання)</b>	<ul style="list-style-type: none"> <li>– аналізувати вимоги до PCOD;</li> <li>– обирати методи та засоби аналізу, моделювання та проектування PCOD відповідно до визначених вимог;</li> <li>– виконувати повний цикл заходів з аналізу, моделювання та проектування PCOD відповідно до визначених вимог;</li> </ul>

	<ul style="list-style-type: none"> <li>– обирати засоби та виконувати повний цикл заходів з конструювання – розробки, РСОД;</li> <li>– забезпечувати відповідність спроектованих моделей та всієї РСОД до реалій компанії, що підлягає інформатизації</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>– моделювати та оптимізувати бізнес-процеси та потоки даних компаній;</li> <li>– моделювати та проектувати РСОД засобами UML;</li> <li>– будувати логічні та фізичні моделі реляційних баз даних;</li> <li>– будувати логічні моделі об'єктних баз даних;</li> <li>– виконувати та використовувати об'єктно-реляційне відображення даних;</li> <li>– маніпулювати даними в централізованих та розподілених базах даних;</li> <li>– створювати та використовувати об'єкти доступу до даних;</li> <li>– розвивати та розширювати функціонал існуючі РСОД;</li> <li>– супроводжувати та трансформувати існуючі РСОД</li> </ul>
<b>Інформаційне забезпечення</b>	<p>Силабус</p> <p>Електронний конспект лекцій:  <a href="http://fpm.kpi.ua/archive/">http://fpm.kpi.ua/archive/</a></p>
<b>Семестровий контроль</b>	Залік

<b>Дисципліна</b>	<b>Відмовостійкі багатопроцесорні системи</b>
<b>Кафедра</b>	Системного програмування і спеціалізованих комп'ютерних систем ФПМ
<b>Рівень ВО</b>	Третій (PhD)
<b>Курс, семестр</b>	2 курс, 4 семестр
<b>Обсягдисципліни</b>	5 кредитів ЄКТС 36 год. аудиторної роботи, 114 год. СРС
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Знання та вміння в межах програми магістра Комп'ютерної інженерії; знання та розуміння принципів, концепцій та технологій побудови обчислювальних систем;
<b>Що буде вивчатися</b>	Методи, засоби та процеси аналізу, моделювання, проектування та конструювання відмовостійких обчислювальних систем (ВОС)
<b>Чому це цікаво/треба вивчати</b>	<p>Питання відмовостійкості обчислювальних систем (fault-tolerant system) стало актуальним з появою потужних обчислювальних систем з розподіленими ресурсами в межах одного комп'ютера (багатоядерні системи), локальних корпоративних і зовнішніх (регіональних і глобальних) мереж, кластерів, технологій пошуку і багатовимірного аналізу даних, розвитком Web-технологій.</p> <p>Особливо воно актуально для бізнесу, де помилка в роботі системи або тимчасової її простої (що становить часом секунди) обертаються більш ніж значними фінансовими втратами.</p> <p>Вкрай сильно від такого роду аварій страждає фінансова індустрія, годину непрацездатного простою якої може обчислюватися мільйонами доларів збитку.</p> <p>Загрозливими для життя і здоров'я людей є помилки в роботі системи управління ядерними об'єктами, хімічними виробництвами, авіаційним рухом і т.п.</p> <p>Таким чином потреба в високонадійних безпечних системах зустрічається там, де мова йде про збої, що тягнуть за собою, катастрофічні наслідки різної природи.</p> <p>Розробка ВОС або її відмовостійка експлуатація, повинні ґрунтуватися на знаннях і розумінні природи тих видів відмов, ймовірність яких, за даних умов експлуатації пристрою, мають максимально велику ймовірність виникнення.</p> <p>Існує два основних напрямки при побудові ВОС. Перший спосіб - використання тільки відмовостійких компонентів, коли кожен компонент системи може продовжувати своє функціонування, навіть якщо один або кілька підкомпонентів системи, виходять з ладу. Другий спосіб це розробка методів, що гарантують побудову ВОС з компонентів, які не є відмовостійкими. У таких системах відмовостійкість реалізована за рахунок введення надмірності і розробки спеціального програмного забезпечення, елементних взаємозв'язків і алгоритмів функціонування.</p> <p>В рамках другого способу розрізняють:</p> <ul style="list-style-type: none"> <li>– апаратну надмірність (Hardware Redundancy) - резервування;</li> <li>– програмну надмірність (Software Redundancy) – обробка одних і тих же даних різними програмами і порівняння їх результатів та виключення спотворення результатів;</li> </ul>

	<ul style="list-style-type: none"> <li>– інформаційну надмірність (InformationRedundancy), коли повідомлення передаються багатократно;</li> <li>– тимчасова надмірність (TimeRedundancy) - використання певної частини продуктивності комп'ютера для контролю за виконанням програм та відновлення обчислювального процесу, наприклад, з подвійним або потрійним прорахунком на обчислювальній машині.</li> </ul> <p>Спеціалісти в галузі аналізу, моделювання, проектування та реалізації ВОС користуються підвищеним попитом на ринку праці, а самі ВОС постійно розвиваються та оновлюються.</p>
<b>Чому можна навчитися (результати навчання)</b>	<ul style="list-style-type: none"> <li>– аналізувати вимоги до ВОС;</li> <li>– обирати методи та засоби аналізу, моделювання та проектування ВОС відповідно до визначених вимог;</li> <li>– виконувати повний цикл заходів з аналізу, моделювання та проектування ВОС відповідно до визначених вимог;</li> <li>– обирати засоби та виконувати повний цикл заходів з конструювання – розробки, ВОС;</li> <li>– забезпечувати відповідність спроектованих моделей та всієї ВОС до реальних компаній та об'єктів, що потребують відмовостійкості обчислювальної системи</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>– моделювати та аналізувати поведінку компаній та об'єктів, що потребують ВОС;</li> <li>– моделювати та проектувати ВОС;</li> <li>– вибирати або комбінувати способи підвищення відмовостійкості систем;</li> <li>– модернізувати ВОС з метою підвищення їх відмовостійкості;</li> <li>– проводити експлуатацію, супроводжувати та трансформувати існуючі ВОС</li> </ul>
<b>Інформаційне забезпечення</b>	Силабус Електронний конспект лекцій: <a href="http://fpm.kpi.ua/archive/">http://fpm.kpi.ua/archive/</a>
<b>Семестровий контроль</b>	Залік

<b>Дисципліна</b>	<b>Децентралізовані застосунки</b>
<b>Кафедра</b>	Програмного забезпечення в енергетиці ННІАТЕ
<b>Рівень ВО</b>	Третій (PhD)
<b>Курс, семестр</b>	2 курс, 4 семестр
<b>Обсяг</b>	5 кредитів ЄКТС 36 год. аудиторної роботи, 114 год. СРС
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Об'єктно-орієнтоване програмування, мережеві технології.
<b>Що буде вивчатися</b>	Технологія Blockchain у дії. Що таке Bitcoin. Що таке децентралізовані застосунки. Підтримувальні технології. Зберігання даних безпосередньо в ланцюжках блоків Bitcoin. Екосистема успішних децентралізованих застосунків. Навіщо і як були створені OpenBazaar і Lighthouse.
<b>Чому це цікаво/треба вивчати</b>	Нині активно формується нова модель створення масштабованих і ефективних застосунків, основи якої були закладені технологією Bitcoin, що включає криптографічну реєстрацію угод, модель з обмеженими ресурсами і пірингові технології. Вказані характеристики послужили відправною точкою для створення нового типу програмного забезпечення, що дістало назву "Децентралізовані застосунки". Децентралізовані застосунки гнучкіше, прозоріше, надійніше і мають більше мотивуючу організацію, ніж сучасне програмне забезпечення, створене із застосуванням традиційних моделей. Це курс, який допоможе зрозуміти ці застосування і навчитися розробляти їх.
<b>Чому можна навчитися (результати навчання)</b>	Реалізація децентралізованої версії Twitter. Механізм BitSwap в IPFS. Децентралізований Twitter. Аналіз проблем безпеки, якими користуються спамери. Використання таких прийомів, як повний цикл доказу виконання роботи при створенні нового користувача для перевірки його ідентичності після реєстрації, для відвертання атаки Сибілі і багато що інше.
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	Скористайтеся цим курсом як трампліном до створення ваших власних застосунків, використовуючих технологію Blockchain.
<b>Інформаційне забезпечення</b>	Силабус Електронний конспект лекцій
<b>Семестровий контроль</b>	Залік

<b>Дисципліна</b>	<b>Математичні методи інформаційних систем</b>
<b>Кафедра</b>	Інформатики та програмної інженерії ФІОТ
<b>Рівень ВО</b>	Третій (PhD)
<b>Курс, семестр</b>	2 курс, 4 семестр
<b>Обсяг</b>	5 кредитів ЄКТС 36 год. аудиторної роботи, 114 год. СРС
<b>Мова викладання</b>	Українська
<b>Вимоги до початку вивчення</b>	Дисципліна “Математичні методи інформаційно-управляючих систем” базується на наступних дисциплінах - “Дискретна математика”, “Теорія ймовірностей”, “Методи і системи штучного інтелекту”, “Моделювання систем”.
<b>Що буде вивчатися</b>	<ul style="list-style-type: none"> <li>– Класифікація, аналіз та прогнозування часових рядів;</li> <li>– аналіз та виявлення аномалій в часових рядах;</li> <li>– основи математичної теорії фракталів, L-системи, генерація фракталів та фрактальних кластерів;</li> <li>– фрактальні часові ряди і фрактальна розмірність.</li> </ul>
<b>Чому це цікаво/треба вивчати</b>	<ul style="list-style-type: none"> <li>– Більшість процесів різної природи описуються часовими рядами і фрактальними часовими рядами безпосередньо;</li> <li>– використання математичних методів для генерації, аналізу і прогнозування динамічних процесів в інформаційно-управляючих системах.</li> </ul>
<b>Чому можна навчитися (результати навчання)</b>	<ul style="list-style-type: none"> <li>– Застосовувати математичні методи аналізу фрактальності часових рядів;</li> <li>– застосовувати математичні методи для прогнозування динамічних процесів різної природи.</li> </ul>
<b>Як можна користуватися набутими знаннями і уміннями (компетентності)</b>	<ul style="list-style-type: none"> <li>– Розробляти програмне забезпечення математичних методів для генерації, аналізу і прогнозування часових рядів динамічних процесів різної природи;</li> <li>– аналізувати і виявляти аномальну поведінку динамічних процесів з використанням фрактальних підходів, лінгвістичного моделювання.</li> </ul>
<b>Інформаційне забезпечення</b>	Силабус Електронний конспект лекцій
<b>Семестровий контроль</b>	Залік