



	<p align="center">Силабус навчальної дисципліни</p> <p align="center">«Технології штучного інтелекту в інженерії програмного забезпечення» Artificial Intelligence Technologies in Software Engineering.</p> <p>Освітньо-наукова програма: Інженерія програмного забезпечення Спеціальність: F2 Інженерія програмного забезпечення Галузь знань: F Інформаційні технології</p>
Рівень вищої освіти	Третій (освітньо-науковий)
Статус дисципліни	Навчальна дисципліна вибіркового компонента вибору фахового переліку
Курс	1 (перший)
Семестр	2 (другий)
Обсяг дисципліни, кредити ЄКТС/години	4 кредити / 120 годин
Мова викладання	Українська
Що буде вивчатися (предмет вивчення)	Теоретичні та архітектурні засади моделювання теорії розуму (Theory of Mind) за допомогою великих мовних моделей (Large Language Models) та їх застосування до ключових процесів розробки програмного забезпечення, де продуктивність команди обмежена розбіжністю ментальних моделей учасників: (i) аналіз вимог в agile-методологіях: застосування абдуктивного виведення (метод TomAbd) для відстеження розбіжності між тим, що заявлено в user story / backlog, і тим, що фактично розуміють стейкхолдери; моделювання переконань при уточненні імог. (Приклад: агент абдукує з коментарів у issue-трекері приховане переконання замовника, яке суперечить формулюванню вимоги, і сигналізує про неї до того, як вона потрапить у код); (ii) тестування: використання абдуктивного ТоМ-механізму для генерації тест-кейсів, що моделюють «хибні переконання» системи про власний стан (аналог задачі Sally-Anne, перенесений на дефекти) (приклад: агент моделює, що саме «знає» модуль про стан зовнішньої залежності (кеш, конфігурацію, версію API) на момент помилки, і абдукує мінімальний сценарій, який відтворює розбіжність між очікуваним і фактичним станом системи на момент тестування; (iii) пояснення семантики програмного забезпечення для LLM — принцип «мова на межі, логіка в ядрі» (LLM shells on multi-context systems): LLM-агент формалізує природномовний опис коду, вимог або дефекту в явні логічні формули (символьне представлення семантики модуля/системи), а символьне ядро виконує міркування,

	<p>ревізію переконань та абдукцію; зворотний напрям — вербалізація логічних висновків назад у природну мову для команди; це дозволяє LLM-асистенту не просто «згадувати» схожий код із тренувальних даних, а явно й аудитовано працювати з переконаннями про поведінку конкретної системи в задачах</p>
<p>Чому це цікаво/треба вивчати (мета)</p>	<p>Курс забезпечує підґрунтя для ознайомлення з підходами до подолання фундаментального обмеження сучасних LLM-асистентів у розробці ПЗ — нездатності послідовно відстежувати розбіжні ментальні моделі учасників команди (розробника, product owner, замовника, тестувальника). Такі розбіжності є прихованими джерелами ризиків на кожному процесі життєвого циклу розробки програмного забезпечення: хибно витлумачені вимоги, неузгоджені архітектурні припущення під час проєктування, дефекти, зумовлені розбіжністю між очікуваним і фактичним станом системи під час тестування. Розглядаються ризики використання LLM для цих задач (втрата контексту переконань, нездатність до аудитованого пояснення рішень) та підхід до їх зниження — поєднання мовних моделей із символічною багатоконтекстною архітектурою, що забезпечує явне, інспектоване представлення переконань команди й підвищує прозорість та надійність AI-асистованих процесів розробки.</p>
<p>Чому можна навчитися (результати навчання)</p>	<p>ПРН01. Мати передові концептуальні та методологічні знання з інженерії програмного забезпечення та дотичних до неї міждисциплінарних напрямів, а також дослідницькі навички, достатні для проведення наукових і прикладних досліджень на рівні останніх світових досягнень з відповідного напрямку, отримання нових знань та/або здійснення інновацій.</p> <p>ПРН03. Пропонувати нові ефективні методи і моделі розроблення, впровадження, супроводу та забезпечення якості програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу.</p> <p>ПРН06. Формулювати і перевіряти гіпотези; використовувати для обґрунтування висновків належні докази, зокрема, результати теоретичного аналізу, експериментальних досліджень і математичного та/або комп'ютерного моделювання, наявні літературні дані.</p>

	<p>ПРН07. Розробляти та досліджувати концептуальні, математичні і комп'ютерні моделі процесів і систем для отримання нових знань та/або створення інноваційних продуктів у інженерії програмного забезпечення та дотичних міждисциплінарних напрямках.</p> <p>ПРН09. Формулювати та вирішувати задачі оптимізації, адаптації, прогнозування, керування та прийняття рішень щодо процесів, засобів та ресурсів розробки, впровадження, супроводу та експлуатації програмного забезпечення.</p>
<p>Як можна користуватися набутими знаннями і уміннями (компетентності)</p>	<p>Загальні компетентності:</p> <p>ЗК01. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.</p> <p>ЗК02. Здатність розв'язувати комплексні проблеми у сфері інженерії програмного забезпечення та з дотичних до неї міждисциплінарних напрямках на основі системного наукового світогляду та загального культурного кругозору із дотриманням принципів професійної етики та академічної доброчесності.</p> <p>ЗК03. Здатність працювати в міжнародному контексті.</p> <p>ЗК04. Здатність презентувати ідеї, інноваційні розробки і результати досліджень як в науковій так і в професійній спільноті.</p> <p>Спеціальні (фахові) компетентності:</p> <p>СК01. Здатність інтегрувати знання з різних галузей, застосовувати системний підхід та враховувати нетехнічні аспекти при розв'язанні комплексних проблем інженерії програмного забезпечення й проведенні досліджень.</p> <p>СК03. Здатність отримувати нові наукові результати, які створюють нові знання та становлять оригінальний внесок у розвиток інженерії програмного забезпечення та дотичних до неї міждисциплінарних напрямів.</p> <p>СК04. Здатність відстежувати тенденції розвитку інженерії програмного забезпечення та критично переосмислювати наявні технології.</p> <p>СК05. Здатність до розроблення нових та вдосконалення існуючих моделей, методів, засобів, процесів у сфері інженерії програмного забезпечення, які забезпечують розвиток або надають нові можливості технологіям розробки та супроводження програмного забезпечення.</p> <p>СК09. Здатність проєктувати, розробляти та впроваджувати інноваційні програмні рішення, що сприяють досягненню цілей сталого розвитку, зокрема забезпеченню якісної освіти (ціль 4), створенню стійкої інфраструктури та інновацій (ціль 9), а також розвитку</p>

	ефективних інституцій (ціль 16), шляхом застосування сучасних методологій розробки програмного забезпечення та інструментів штучного інтелекту.
Навчальна логістика	Види занять: лекція, практичне. Форми навчання: очна, дистанційна. Методи навчання: загальнонаукові та спеціальні методи: 1) методи організації та здійснення навчально-пізнавальної діяльності: словесні методи навчання: лекція, бесіда, пояснення; 2) наочні методи навчання: ілюстрування, демонстрування; 3) практичні методи навчання: практична робота та дослідна робота; 4) методи стимулювання навчальної діяльності: методи навчальної дискусії, метод опори на життєвий досвід студентів; 5) метод контролю і самоконтролю у навчанні: метод усного і письмового контролів.
Пререквізити	Дисципліни «Інформаційне забезпечення наукових досліджень» та «Системно-синергетичне моделювання об'єктів досліджень та математичні методи обробки даних за спеціальністю інженерія програмного забезпечення» та «Академічна англійська мова та наукове письмо».
Інформаційне забезпечення з репозитарію та фонду НТБ НАУ	Нікольський, Ю. В. Системи штучного інтелекту [Текст] : навчальний посібник / Ю. В. Нікольський, В. В. Пасічник, Юрій Миколайович Щербина ; МОН МС України. – 2-е вид., випр. та доп. – Львів : Магнолія-2006, 2015. – 279 с. – (Комп'ютинг). Тарановський, А. О. Створення основних типів тестових завдань за допомогою штучного інтелекту [Текст] / А. О. Тарановський, В. Д. Самойлов // Електронне моделювання : Міжнародний науково-теоретичний журнал / Інститут проблем моделювання в енергетиці ім. Г.Є. Пухова НАН України. – Київ : Академперіодика, 2024. – № 2. – С. 88–100.
Локація та матеріально-технічне забезпечення	Технічні засоби індивідуального підключення до google класу
Семестровий контроль, екзаменаційна методика	Модульна контрольна робота, диф. залік
Кафедра	Кафедра інженерії програмного забезпечення
Факультет	Факультет комп'ютерних наук та технологій
Викладач(і)	ПІБ викладача: Олена Вікторівна Чебанюк Посада: професор Науковий ступінь: д.т.н.

	Вчене звання: професор Профайл викладача: https://www.scopus.com/authid/detail.uri?authorId=57203264737 Телефон: (044) 406-70-98 E-mail: olena.chebaniuk@npp.kai.edu.ua Робоче місце: 6.309
Оригінальність навчальної дисципліни	Авторський курс, викладання українською мовою.
Лінк на дисципліну	Після формування групи слухачів створюється кабінет у Google Classroom із необхідними навчальними матеріалами.
Максимальна кількість слухачів	50