

Затверджено
завідувач кафедри КІ
 Ігор БУЧЕНКО
“ 31 ” серпня 2026 р.

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «ТЕОРІЯ І ТЕХНОЛОГІЇ РОЗРОБКИ ПРОГРАМНИХ СИСТЕМ»

Лектор курсу			Худік Богдан Олександрович, доктор філософії, доцент кафедри Інженерії програмного забезпечення		Контактна інформація лектора, сторінка курсу в Google classroom	e-mail: b.khudik@duikt.edu.ua сторінка курсу в Google Classroom https://classroom.google.com/c/ODM2MTAyOTY5N_DM2?cjc=72jluzjv	
Галузь знань			F Інформаційні технології		Рівень вищої освіти	доктор філософії	
Спеціальність			F7 Комп'ютерна інженерія		Семестр	2	
Освітня програма			Комп'ютерні науки		Тип дисципліни	Цикл обов'язкових дисциплін ОНП	
Обсяг :	Кредитів ECTS	Годин	За видами занять:				
			Лекцій	Семінарських занять	Практичних занять	Лабораторних занять	Самостійна підготовка
	3	90	18	-	18	-	54
АНОТАЦІЯ КУРСУ							
Освітні компоненти, які передують вивченню			Організація науки та наукових досліджень, Сучасні методи викладання у вищій школі, Філософія сучасної науки				
Освітні компоненти, для яких є базовою			Концепції розвитку програмно-апаратних платформ для корпоративної ІТ-інфраструктури, Концепції розвитку штучного інтелекту				
Мета курсу:	Сформувати у аспірантів знання та вміння, дослідницькі навички, необхідні для створення нових засобів розробки програмних систем на основі сучасного математичного апарату.						
Компетентності відповідно до освітньої програми							
Загальні компетентності (ЗК)					Спеціальні (фахові) компетентності (СК)		
ЗК01. Здатність до абстрактного мислення, аналізу та синтезу. ЗК02. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.					СК01. Здатність виконувати оригінальні дослідження, досягати наукових результатів, які створюють нові знання у комп'ютерній інженерії та дотичних до неї міждисциплінарних напрямках і		

ЗК04. Здатність розв'язувати комплексні проблеми у сфері комп'ютерної інженерії на основі системного наукового світогляду та загального культурного кругозору із дотриманням принципів професійної етики та академічної доброчесності.	можуть бути опубліковані у провідних наукових виданнях з комп'ютерної інженерії та суміжних галузей.
--	--

Програмні результати навчання (РН)

ПРН1. Мати передові концептуальні та методологічні знання об'єктів професійної діяльності комп'ютерної інженерії і на межі предметних галузей, а також дослідницькі навички, достатні для проведення наукових і прикладних досліджень на рівні останніх світових досягнень з комп'ютерної інженерії, ІТ-інфраструктур та інформаційних технологій, отримання нових знань та/або здійснення інновацій.

ПРН 2. Знати сучасні методи проведення досліджень в галузі комп'ютерної інженерії та інформаційних технологій, а саме: способи подання, отримання, зберігання, передавання, опрацювання та захисту інформації, математичні моделі обчислювальних процесів, технології виконання обчислень (високопродуктивних, паралельних, розподілених, мобільних, веб-базованих та хмарних, зелених або енергоефективних, безпечних, автономних, адаптивних, інтелектуальних), а також квантових, біомолекулярних, оптичних та оброблення великих даних тощо, а також технології людино-машинної взаємодії та кооперації, доданої та віртуальної реальності.

ПРН4. Знати і розуміти наукові і математичні положення, що лежать в основі функціонування програмних, програмовних і програмно-технічних комп'ютерних засобів, систем та мереж, Інтернету речей, систем для оброблення великих даних.

ПРН6. Вміти ефективно здійснювати пошук та критичний аналіз інформації з різних джерел.

ПРН7. Вміти розв'язувати задачі синтезу та аналізу об'єктів дослідження комп'ютерної інженерії та їх окремих складових серед яких: аналогові та цифрові комп'ютери (електронні, квантові, біомолекулярні, оптичні тощо) та комп'ютерні системи універсального або спеціального призначення (стаціонарні, мобільні, вбудовані, розподілені тощо); локальні, глобальні комп'ютерні мережі; кіберфізичні системи, Інтернет речей, системи для оброблення великих даних та штучного інтелекту, ІТ-інфраструктури; їх програмно-технічні засоби (апаратні, програмні, програмовані, реконфігуровані, системне та прикладне програмне забезпечення), інтерфейси та протоколи взаємодії їх компонентів.

ПРН8. Вміти розробляти та досліджувати концептуальні, математичні і комп'ютерні моделі, інформаційні процеси, технології, методи, способи, інструментальні засоби та системи для автоматизованого та автоматичного проектування; налагодження, виробництва й експлуатації комп'ютерів та комп'ютерних систем і мереж, кіберфізичних систем, Інтернету речей та ІТ-інфраструктур, розроблення, верифікації та розгортання програмного забезпечення та систем у хмарних та інших середовищах, забезпечення якості, надійності та безпеки а також ефективно використовувати їх для отримання нових знань та/або створення інноваційних продуктів у комп'ютерній інженерії та дотичних міждисциплінарних напрямках.

ОРГАНІЗАЦІЯ НАВЧАННЯ

Тема, опис теми	Ви д занят тя	Оцінюва ння за тему	Форми і методи навчання/питання до самостійної роботи
Змістовний модуль 1. Теоретичні засади деяких питань розробки програмних систем			
Тема 1. Особливості виконання вимог концепцій Coupling та Cohesion при розробці програмних систем.			
Знати: мету та зміст вимог концепцій Coupling та Cohesion при розробці програмних систем.			
Вміти: обирати архітектурні рішення для виконання вимог концепцій Coupling та Cohesion при розробці різноманітних програмних систем.			
Формування компетенцій: ЗК 1, ЗК 2, ЗК 4, СК07.			

Результати навчання: РН12 Рекомендовані джерела: 1, 2.			
Заняття 1.1. Концепції Coupling та Cohesion.	Лекція 1 2 год		Пояснювально-ілюстративний, лекція-бесіда, бліц-опитування
Заняття 1.2. Реалізація класів та методів для виконання вимог концепцій Low Coupling та High Cohesion.	Практичне заняття 1 2 год	4 бали	Колаборативне навчання, кейс-метод, мозгова атака, усне опитування
Тема 2. Порівняльний аналіз використання перевизначених методів підкласів типу Model та статичних методів класів типу Controller. Знати: особливості використання перевизначених та статичних методів. Вміти: створювати методи обидвох типів та в залежності від завдання програмних систем обирати більш ефективний спосіб реалізації методів. Формування компетенцій: ЗК 1, ЗК 2, ЗК 4, СК07. Програмні результати: РН12. Рекомендовані джерела: 1, 2.			
Заняття 2.1. Порівняльний аналіз використання перевизначених методів підкласів типу Model та статичних методів класів типу Controller.	Лекція 2 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування
Заняття 2.2. Порівняння ефективності перевизначених та статичних методів в різних умовах, для реалізації однакових завдань.	Практичне заняття 2 2 год	4 бали	Колаборативне навчання, демонстрування, кейс-метод, робота над помилками
Тема 3. Порівняльні властивості функціональних та класичних інтерфейсів. Знати: властивості класичних інтерфейсів та стандартних функціональних інтерфейсів пакету java.util.function. Вміти: реалізувати множинне наслідування на базі класичних інтерфейсів та анонімних внутрішніх класів, а також формувати складні аргументи для методів на основі стандартних функціональних інтерфейсів. Формування компетенцій: ЗК 1, ЗК 2, ЗК 4, СК07. Програмні результати: РН12. Рекомендовані джерела: 1, 2.			
Заняття 3.1. Порівняльні властивості функціональних та класичних інтерфейсів.	Лекція 3 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування
Заняття 3.2. Формування складних аргументів для методів на основі стандартних функціональних інтерфейсів.	Практичне заняття 3 2 год	4 бали	Колаборативне навчання, демонстрування, кейс-метод, робота над помилками
Тема 4. Архітектурні особливості розміщення в пам'яті об'єктів при серіалізації/десеріалізації. Знати: синтаксичні вимоги використання методів серіалізації/десеріалізації інтерфейсів Serializable/Externalizable. Вміти: використовувати архітектурні особливості формування конструкторами десеріалізованих об'єктів класів, наслідуваних від несеріалізованих суперкласів.			

Формування компетенцій: ЗК 1, ЗК 2, ЗК 4, СК07. Програмні результати: РН12. Рекомендовані джерела: 1, 2.			
Заняття 4.1. Архітектурні особливості розміщення в пам'яті об'єктів при серіалізації/десеріалізації.	Лекція 4 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування
Заняття 4.2. Серіалізація/десеріалізація об'єктів класів із несеріалізованими полями та несеріалізованими суперкласами.	Практичне заняття 4 2 год	4 бали	Колаборативне навчання, демонстрування, кейс-метод, робота над помилками
Тема 5. Особливості застосування технології Generics для роботи з різними типами контейнерів даних. Знати: принципи безпеки технології Generics. Вміти: реалізувати безпечні спискові контейнери даних у випадку legacy-кодів та необхідності змін даних у контейнерах, а також синхронізовані Maps. Формування компетенцій: ЗК 1, ЗК 2, ЗК 4, СК07. Програмні результати: РН12. Рекомендовані джерела: 1, 2.			
Заняття 5.1. Принципи безпеки технології Generics.	Лекція 5 2 год		Пояснювально-ілюстративний, лекція-бесіда, бліц опитування
Заняття 5.2. Дослідження ефективності використання узагальнених Maps, синхронізованих засобами пакетів java.util та java.util.concurrent.	Практичне заняття 5 2 год	4 бали	Колаборативне навчання, кейс-метод, мозгова атака, усне опитування
	Самостійна робота		
Тема 1. Особливості виконання вимог концепцій Coupling та Cohesion при розробці програмних систем.	6 год	2 бали	Робота в IDE Eclipse.
Тема 2. Порівняльний аналіз використання перевизначених методів підкласів типу Model та статичних методів класів типу Controller.	6 год	2 бали	Робота в IDE Eclipse.
Тема 3. Порівняльні властивості функціональних та класичних інтерфейсів.	6 год	2 бали	Робота в IDE Eclipse.
Тема 4. Архітектурні особливості розміщення в пам'яті об'єктів при серіалізації/десеріалізації.	6 год	3 бали	Робота в IDE Eclipse.
Тема 5. Особливості застосування технології Generics для роботи з різними типами контейнерів даних.	6 год	3 бали	Робота в IDE Eclipse.

Змістовний модуль 2. Технології розробки програмних систем

Тема 6. Технології обробки *Exceptions*.

Знати: особливості виникнення контрольованих/неконтрольованих винятків, обробки винятків або її передавання для різних завдань в програмних системах.

Вміти: ефективно обирати спосіб обробки винятків в залежності від завдання в програмних системах.

Формування компетенцій: ЗК 2, ЗК 3, ЗК 4, ЗК 5, СК07.

Результати навчання: РН12.

Рекомендовані джерела: 1, 2.

Заняття 6.1. Особливості виникнення, обробки та генерування <i>Exceptions</i> .	Лекція 6 2 год		Пояснювально-ілюстративний, лекція-візуалізація, бліц опитування
Заняття 6.2. Принципи ефективної роботи з неконтрольованими/контрольованими винятками.	Практичне заняття 6 2 год	4 бали	Колаборативне навчання, демонстрування, мозгова атака, робота над помилками

Тема 7. Технології обробки текстових даних (*parsing*).

Знати: Алгоритмічні основи обробки текстових даних.

Вміти: формувати та використовувати необхідні регулярні вирази для рішення завдань в програмних системах.

Формування компетенцій: ЗК 2, ЗК 3, ЗК 4, ЗК 5, СК07.

Результати навчання: РН12.

Рекомендовані джерела: 1, 2.

Заняття 7.1. Алгоритмічні основи обробки текстових даних.	Лекція 7 2 год		Пояснювально-ілюстративний, лекція-бесіда, бліц опитування
Заняття 7.2. Формування та порівняльний аналіз регулярних виразів для пошуку лексем та для розкладання великих текстових даних.	Практичне заняття 7 2 год	4 бали	Колаборативне навчання, демонстрування, мозгова атака, робота над помилками

Тема 8. Технології роботи в синхронних багатопотокових програмних системах (*threads*).

Знати: принципи синхронізації в багатопотокових (multithreads) програмних системах.

Вміти: обирати ефективні способи обробки даних в синхронних багатопотокових (multithreads) програмних системах.

Формування компетенцій: ЗК 2, ЗК 3, ЗК 4, ЗК 5, СК07.

Результати навчання: РН12.

Рекомендовані джерела: 3.

Заняття 8.1. Класифікація та порівняльний аналіз синхронних багатопотокових (multithreads) програмних систем.	Лекція 8 2 год		Пояснювально-ілюстративний, лекція-бесіда, бліц опитування
Заняття 8.2. Аналіз обмежень синхронних багатопотокових (multithreads) програмних систем на базі пакетів java.util.concurrent та java.util.concurrent.	Практичне заняття 8 2 год	4 бали	Колаборативне навчання, демонстрування, мозгова атака, робота над помилками

Тема 9. Технології роботи в синхронних паралельних системах (Parallel Streams).

Знати: принципи синхронізації в багатопотокових програмних системах.

Вміти: використовувати засоби стандартних пакетів API для багатопотокових паралельних систем та формувати користувацькі синхронні кодові блоки.

Формування компетенцій: ЗК 2, ЗК 3, ЗК 4, ЗК 5, СК07.

Результати навчання: РН12.

Рекомендовані джерела: 3.

Заняття 9.1. Технології роботи в синхронних паралельних системах (Parallel Streams).	Лекція 9 2 год		Пояснювально-ілюстративний, лекція-бесіда, бліц опитування
Заняття 9.2. Принципи побудови синхронних програмних систем із сумісних використанням багатопотокової (multithreads) та паралельної (Parallel Streams) обробки потоків даних.	Практичне заняття 9 2 год	4 бали	Колаборативне навчання, кейс-метод, мозгова атака, усне опитування.
Самостійна робота			
Тема 6. Технології обробки Exceptions.	6 год	3 бали	Робота в IDE Eclipse.
Тема 7. Технології обробки текстових даних (parsing).	6 год	3 бали	Робота в IDE Eclipse.
Тема 8. Технології роботи в синхронних багатопотокових програмних системах (threads).	6 год	3 бали	Робота в IDE Eclipse.
Тема 9. Технології роботи в синхронних паралельних системах (Parallel Streams).	6 год	3 бали	Робота в IDE Eclipse.
МАТЕРІАЛЬНО-ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ ДИСЦИПЛІНИ			
- Мультимедійний проектор, інтерактивна дошка та/або віртуальна дошка. - Комп'ютерний клас для проведення практичних та лабораторних занять із встановленим програмним забезпеченням Eclipse та доступом до мережі Internet.			
ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ ДИСЦИПЛІНИ			
- Aaron Ploetz. Code with Java 21. A Practical Approach for Building Robust and Efficient Applications.– BPB Publications India, 2024. – 407 p. - Benjamin J.Evans, Jason Clark, Martin Verburg. The Well-Grounded Java Developer. Second Ed. – Manning Publications, 2022. – 705 p. - Java Concurrency in Practice. Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes, Doug Lea. – Addison-Wesley, 2023. – 459 p.			
ПОЛІТИКА КУРСУ («ПРАВИЛА ГРИ»)			
- Курс передбачає роботу в колективі. - Середовище в аудиторії є дружнім, творчим, відкритим до конструктивної критики. - Освоєння дисципліни передбачає обов'язкове відвідування лекцій і практичних занять, а також самостійну роботу. - Самостійна робота включає в себе теоретичне вивчення питань, що стосуються тем лекційних занять, які не ввійшли в теоретичний курс, або ж були розглянуті коротко, їх поглиблена проробка за рекомендованою літературою. - Усі завдання, передбачені програмою, мають бути виконані у встановлений термін. - Якщо студент відсутній з поважної причини, він презентує виконані завдання під час самостійної підготовки та консультації викладача. - Під час роботи над завданнями не допустимо порушення академічної доброчесності: при використанні Інтернет ресурсів та інших джерел інформації студент повинен вказати джерело, використане в ході виконання завдання. Виявлення ознак академічної не доброчесності в практичній (письмовій) роботі студента є підставою для її не зарахування викладачем.			

- Студент, який спізнився має право бути присутнім на занятті. Студенти мають інформувати старосту про неможливість відвідати заняття.
- Користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття в цілях не пов'язаних з навчанням є підставою для не зарахування викладачем роботи студента.

КРИТЕРІЇ ТА МЕТОДИ ОЦІНЮВАННЯ

Умовою допуску до підсумкового контролю є виконання всіх практичних робіт і виконання самостійних завдань, які передбачені структурою освітньої компоненти Теорія і технології розробки програмних систем.

Якщо аспіранта не допущено до складання іспиту, як такого, що не виконав індивідуальний план, йому надається час до перескладання для виконання всіх вимог допуску. Аспірант має право на два перескладання. При повторному перескладанні іспиту його у аспіранта може приймати комісія, яка створюється директором ННІТ. Оцінка комісії є остаточною. У випадку отримання аспірантом 0 балів (неприйнятно), що тягне відрахування за невиконання навчального плану.

Оцінювання аспірантів здійснюється за накопичувальною 100-бальною системою і складається із двох основних оцінкових блоків і розподіляється в певних пропорціях 60 (бали напрацьовані під час вивчення дисципліни – Поточний контроль), 40 (підсумкове оцінювання - Іспит):

Форми контролю	Види навчальної роботи	Оцінювання
ПОТОЧНИЙ	Виконання практичних робіт	36 балів
	Самостійна робота	24 бали
ПІДСУМКОВЕ ОЦІНЮВАННЯ Іспит	Метою іспиту є контроль сформованості практичних навичок та професійних компетентностей, необхідних для виконання професійних обов'язків. Іспит проходить у письмовій формі.	40 балів
Додаткова оцінка		
Види навчальної роботи		Оцінювання
Участь у наукових конференціях, підготовка наукових публікацій за тематикою освітньої компоненти:		
- тези доповіді фахового спрямування;		3 бали
- стаття у фаховому виданні;		5 балів
- стаття в іноземному рецензованому виданні.		10 балів
Максимальна кількість додаткових балів, які можуть бути зараховані здобувачу освіти - 18 балів.		

ПІДСУМКОВА ОЦІНКА ЗА ДИСЦИПЛІНУ

бали	Критерії оцінювання	Рівень компетентності	Оцінка /запис в екзаменаційній відомості
	Студент демонструє повні й міцні знання навчального матеріалу в обсязі, що відповідає робочій програмі дисципліни, правильно й обґрунтовано приймає необхідні рішення в різних нестандартних ситуаціях. Вміє реалізувати теоретичні положення дисципліни в практичних розрахунках, аналізувати та порівнювати дані об'єктів діяльності фахівця на основі набутих з даної та суміжних дисциплін	Високий Повністю забезпечує вимоги до знань, умінь і навичок, що викладені в робочій програмі дисципліни. Власні пропозиції студента в оцінках і	

90-100	знань та умінь. Знає сучасні технології та методи розрахунків з даної дисципліни. За час навчання при проведенні практичних занять, виконанні індивідуальних завдань проявив вміння самостійно вирішувати поставлені задачі, активно включатись в дискусії, може відстоювати власну позицію в питаннях та рішеннях, які розглядаються. Зменшення 100-бальної оцінки може бути пов'язане з недостатнім розкриттям питань, що стосується тем дисципліни, але виходить за рамки об'єму матеріалу, передбаченого робочою програмою, або студент проявляє невпевненість в тлумаченні теоретичних положень або складних завдань.	вирішенні практичних задач підвищує його вміння використовувати знання, які він отримав при вивченні інших дисциплін, а також знання, набуті при самостійному поглибленому вивченні питань, за тематикою дисципліни, яка вивчається.	Відмінно / Зараховано (А)
---------------	---	---	----------------------------------

82-89	Студент демонструє гарні знання, добре володіє матеріалом, що відповідає робочій програмі дисципліни, робить на їх основі аналіз можливих ситуацій та вміє застосовувати теоретичні положення при вирішенні практичних задач, але допускає окремі неточності. Вміє самостійно виправляти допущені помилки, кількість яких є незначною. Знає сучасні технології та методи розрахунків з даної дисципліни. За час навчання при проведенні практичних занять, виконанні індивідуальних завдань та поясненні прийнятих рішень, дає вичерпні пояснення.	Достатній Забезпечує студенту самостійне вирішення основних практичних задач в умовах, коли вихідні дані в них змінюються порівняно з прикладами, що розглянуті при вивченні дисципліни	Добре / Зараховано (В)
75-81	Студент в загальному добре володіє матеріалом, знає основні положення матеріалу, що відповідає робочій програмі дисципліни, робить на їх основі аналіз можливих ситуацій та вміє застосовувати при вирішенні типових практичних завдань, але допускає окремі неточності. Вміє пояснити основні положення виконаних завдань та дати правильні відповіді при зміні результату при заданій зміні вихідних параметрів. Помилки у відповідях / рішеннях / розрахунках не є системними. Знає характеристики основних положень, які мають визначальне значення при проведенні практичних занять, виконанні індивідуальних завдань та поясненні прийнятих рішень дисципліни, що вивчається.	Достатній Конкретний рівень, за вивченим матеріалом робочої програми дисципліни. Додаткові питання про можливість використання теоретичних положень для практичного використання викликають утруднення.	Добре / Зараховано (С)
67-74	Студент засвоїв основний теоретичний матеріал, передбачений робочою програмою дисципліни, та розуміє постанову стандартних практичних завдань, має пропозиції щодо напрямку їх вирішень. Розуміє основні положення, що є визначальними в курсі, може вирішувати подібні завдання тим, що розглядались з викладачем, але допускає значну кількість незначних і грубих помилок, які може усувати за допомогою викладача.	Середній Забезпечує достатньо надійний рівень відтворення основних положень дисципліни	Задовільно / Зараховано (D)
60-66	Студент має певні знання, передбачені в робочій програмі дисципліни, володіє основними положеннями, що вивчаються на рівні, який визначається як мінімально допустимий. З використанням основних теоретичних положень, студент з труднощами пояснює правила вирішення практичних / розрахункових завдань дисципліни. Виконання практичних / індивідуальних завдань	Середній Є мінімально допустимим у всіх складових навчальної програми з дисципліни	Задовільно / Зараховано (Е)

	значно формалізовано: є відповідність алгоритму, але відсутнє глибоке розуміння роботи та взаємозв'язків з іншими дисциплінами.		
35-59	Студент може відтворити окремі фрагменти з курсу. Незважаючи на те, що програму навчальної дисципліни студент виконав, працював він пасивно, його відповіді під час практичних робіт в більшості є невірними, необґрунтованими. Цілісність розуміння матеріалу з дисципліни у студента відсутні.	Низький Не забезпечує практичної реалізації задач, що формуються при вивченні дисципліни	Незадовільно з можливістю повторного складання) / Не зараховано (FX) В залікову книжку не представляється
0-34	Студент повністю не виконав вимог робочої програми навчальної дисципліни. Його знання на підсумкових етапах навчання є фрагментарними. Студент не допущений до здачі заліку.	Незадовільний Студент не підготовлений до самостійного вирішення задач, які окреслює мета та завдання дисципліни	Незадовільно з обов'язковим повторним вивченням / Не допущений (F) В залікову книжку не представляється

ПОЛІТИКА ДОБРОЧЕСНОСТІ

Здобувач вищої освіти виконуючи самостійну або індивідуальну роботу повинен дотримуватись політики доброчесності. У разі наявності плагіату в будь-яких видах робіт Здобувача він отримує незадовільну оцінку і повинен повторно виконати завдання, які передбачені у Силабусі.