

Чернівецький національний університет імені Юрія Федьковича

Факультет математики та інформатики

Кафедра прикладної математики та інформаційних технологій

“ЗАТВЕРДЖУЮ”

Декан факультету математики та інформатики

_____ Ольга МАРТИНЮК

«___» _____ 2026 року

РОБОЧА ПРОГРАМА

навчальної дисципліни

Алгоритми і структури даних

Обов’язкова навчальна дисципліна

Освітньо-професійна програма	<u>Технології програмування та комп’ютерне моделювання</u>
Спеціальність	<u>F1 Прикладна математика</u>
Галузь знань	<u>F Інформаційні технології</u>
Рівень вищої освіти	<u>перший (бакалаврський)</u>
Факультет	<u>математики та інформатики</u>
Мова навчання	<u>українська</u>

Чернівці, 2026 рік

Робоча програма навчальної дисципліни "Алгоритми і структури даних" **складена відповідно до** освітньо-професійної програми "Технології програмування та комп'ютерне моделювання" першого (бакалаврського) рівня вищої освіти за спеціальністю "F1 Прикладна математика" галузі знань "F Інформаційні технології".

Розробник:

Краснокутська Інесса Володимирівна, доцент кафедри прикладної математики та інформаційних технологій, кандидат фізико-математичних наук, доцент

Викладач, що забезпечує читання даної навчальної дисципліни:

Краснокутська Інесса Володимирівна, доцент кафедри прикладної математики та інформаційних технологій, кандидат фізико-математичних наук, доцент

Погоджено з гарантом ОП _____ **Василь МАЦЕНКО**

Затверджено на засіданні кафедри прикладної математики та інформаційних технологій

Протокол № ____ від «__» _____ 2026 року

Завідувач кафедри _____ **Ярослав БІГУН**

Схвалено методичною радою факультету математики та інформатики

Протокол № ____ від «__» _____ 2026 року

Голова методичної ради _____ **Тоня ФРАТАВЧАН**

МЕТА ОСВІТНЬОГО КОМПОНЕНТА

Мета дисципліни:

- формування у студентів системного уявлення про основні алгоритми та структури даних, їх властивості, переваги та області застосування;
- оволодіння теоретичними засадами побудови та аналізу алгоритмів, оцінювання їх ефективності за часовими та просторовими характеристиками;
- набуття практичних навичок використання лінійних і нелінійних структур даних для розв'язування прикладних задач;
- розвиток алгоритмічного мислення та вміння обирати оптимальні структури даних і алгоритмічні підходи залежно від умов задачі.

Завдання дисципліни:

- вивчення базових структур даних, зокрема списків, списків з ітераторами, однозв'язних, двозв'язних та кільцевих списків;
- засвоєння принципів роботи та практичного застосування стеків і черг;
- вивчення деревоподібних структур даних та основних операцій над ними;
- опанування алгоритмів сортування та пошуку, аналіз їх складності та ефективності;
- вивчення алгоритмічної стратегії «розділяй та володарюй» та її використання для побудови ефективних алгоритмів;
- формування навичок реалізації алгоритмів і структур даних мовами програмування з використанням сучасних підходів та інструментів.

Знання, уміння та навички, отримані в результаті вивчення дисципліни, є фундаментальними для підготовки фахівця у галузі інформаційних технологій, сприяють розвитку логічного та алгоритмічного мислення й становлять основу для подальшого вивчення складніших програмних та обчислювальних дисциплін.

ПРЕРЕКВІЗИТИ

Для успішного засвоєння дисципліни необхідні добрі знання класичного програмування мовами C, C++, Python, що закладаються під час навчання в бакалавраті в навчальних курсах, пов'язаних з вивченням мов програмування.

РЕЗУЛЬТАТИ НАВЧАННЯ

В результаті вивчення дисципліни студент має набути таких **компетентностей**:

знати:

- основні поняття теорії алгоритмів та структур даних;
- класифікацію та властивості алгоритмів, методи оцінювання їх часової та просторової складності;

- лінійні структури даних: списки, списки з ітераторами, однозв'язні, двозв'язні та кільцеві списки;
- принципи роботи стеків і черг та сфери їх застосування;
- деревоподібні структури даних та основні операції над ними;
- алгоритми сортування та пошуку, їх переваги, недоліки та області використання;
- алгоритмічну стратегію «розділяй та володарюй» та приклади її застосування;
- основні підходи до організації та зберігання даних у пам'яті комп'ютера.

вміти:

- аналізувати поставлені задачі та обирати адекватні алгоритми і структури даних для їх розв'язання;
- реалізовувати лінійні та нелінійні структури даних мовами програмування;
- застосовувати ітератори для обходу та модифікації структур даних;
- використовувати стеки та черги для розв'язування прикладних задач;
- будувати та опрацьовувати дерева, виконувати основні операції над ними;
- реалізовувати та порівнювати алгоритми сортування і пошуку;
- застосовувати підхід «розділяй та володарюй» при розробці ефективних алгоритмів;
- оцінювати ефективність алгоритмів і оптимізувати їх за часовими та просторовими характеристиками.

Під час вивчення дисципліни, відповідно до освітньо-професійної програми, формуються наступні

загальні компетентності:

ЗК01. Здатність учитися і оволодівати сучасними знаннями.

ЗК06. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК07. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК10. Навички у використанні інформаційних і комунікаційних технологій.

фахові компетентності:

ФК02. Здатність виконувати завдання, сформульовані у математичній формі.

ФК04. Здатність розробляти алгоритми та структури даних, програмні засоби та програмну документацію.

ФК06. Здатність розв'язувати професійні задачі за допомогою комп'ютерної техніки, комп'ютерних мереж та Інтернету, в середовищі сучасних операційних систем, з використанням стандартних офісних додатків.

ФК08. Здатність використовувати сучасні технології програмування та тестування програмного забезпечення.

ФК09. Здатність до проведення математичного і комп'ютерного моделювання, аналізу та обробки даних, обчислювального експерименту, розв'язання формалізованих задач за допомогою спеціалізованих програмних засобів.

та отримуються наступні **програмні результати навчання:**

ПРН04. Виконувати математичний опис, аналіз та синтез дискретних об'єктів та систем, використовуючи поняття й методи дискретної математики та теорії алгоритмів.

ПРН09. Будувати ефективні щодо точності обчислень, стійкості, швидкодії та витрат системних ресурсів алгоритми для чисельного дослідження математичних моделей та розв'язання практичних задач.

ПРН10. Володіти методиками вибору раціональних методів та алгоритмів розв'язання математичних задач оптимізації, дослідження операцій, оптимального керування і прийняття рішень, аналізу даних.

ПРН11. Вміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символьних алгоритмів.

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Загальна інформація

Форма навчання	Рік підготовки	Семестр	Кількість		Кількість годин						Вид підсумкового контролю
			кредитів	годин	лекції	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	2	1	4	120	30	-	-	30	60	-	іспит
Заочна	-	-	-	-	-	-	-	-	-	-	-

Структура змісту навчальної дисципліни

Назви змістових модулів і тем	Кількість годин											
	денна форма						Заочна форма					
	усього	у тому числі										
		л	п	лаб	інд	с.р.						
1	2	3	4	5	6	7						
Змістовий модуль 1. Динамічні структури даних												
Тема 1. Однозв’язні та двоз’язні списки	12	2		2		8						
Тема 2. Узагальнене програмування. Ітератори. Списки з ітераторами	12	2		2		8						
Тема 3. Стеки, черги та їх застосування. Обернена польська нотація	16	4		4		8						
Тема 4. Кільцеві списки. Кільцеві списки з ітераторами. Деки	16	4		4		8						
Разом за змістовим модулем 1	56	12		12		32						
Змістовий модуль 2. Алгоритми та методи розробки алгоритмів												
Тема 1.	6	2		2		2						

Асимптотичний аналіз. Оцінка складності роботи алгоритму												
Тема 2. Алгоритми пошуку	10	2		2		6						
Тема 3. Алгоритми сортування	14	4		4		6						
Тема 4. Метод декомпозиції (розділяй та володарюй)	14	4		4		6						
Разом за змістовим модулем 2	44	12		12		20						
Змістовий модуль 3. Складніші структури даних та методи розробки алгоритмів												
Тема 1. Древа. Бінарні дерева. Бінарні дерева пошуку	6	2		2		2						
Тема 2. Піраміди. Черги з пріоритетами	6	2		2		2						
Тема 3. Backtracking (пошук з поверненням)	4	1		1		2						
Тема 4. Динамічне програмування. Жадібні алгоритми	4	1		1		2						
Разом за змістовим модулем 3	20	6		6		8						
Усього годин	120	30		30		60						

Тематика лекційних занять

МОДУЛЬ 1. ДИНАМІЧНІ СТРУКТУРИ ДАНИХ	
Тема 1	Однозв'язні та двозв'язні списки
Тема 2	Узагальнене програмування. Ітератори. Списки з ітераторами
Тема 3	Стеки, черги та їх застосування. Обернена польська нотація.
Тема 4	Кільцеві списки. Кільцеві списки з ітераторами. Деки
МОДУЛЬ 2. АЛГОРИТМИ ТА МЕТОДИ РОЗРОБКИ АЛГОРИТМІВ	
Тема 1	Асимптотичний аналіз. Оцінка складності роботи алгоритму

Тема 2	Алгоритми пошуку
Тема 3	Алгоритми сортування
Тема 4	Метод декомпозиції (розділяй та володарюй)
МОДУЛЬ 3. СКЛАДНІШІ СТРУКТУРИ ДАНИХ ТА МЕТОДИ РОЗРОБКИ АЛГОРИТМІВ	
Тема 1	Дерева. Бінарні дерева. Бінарні дерева пошуку
Тема 2	Піраміди. Черги з пріоритетами
Тема 3	Backtracking (пошук з поверненням)
Тема 4	Динамічне програмування. Жадібні алгоритми

Тематика семінарських занять

Семінарські заняття з даної навчальної дисципліни навчальним планом не передбачені.

Тематика практичних занять

Практичні заняття з даної навчальної дисципліни навчальним планом не передбачені.

Тематика лабораторних занять

МОДУЛЬ 1. ДИНАМІЧНІ СТРУКТУРИ ДАНИХ	
ЛР 1.1	Однозв'язні та двозв'язні списки
ЛР 1.2	Списки з ітераторами
ЛР 2.1	Стеки, черги. Обернена польська нотація.
ЛР 2.2	Кільцеві списки. Кільцеві списки з ітераторами
МОДУЛЬ 2. АЛГОРИТМИ ТА МЕТОДИ РОЗРОБКИ АЛГОРИТМІВ	
ЛР 3	Алгоритми сортування
ЛР 4	Метод декомпозиції (розділяй та володарюй)
МОДУЛЬ 3. СКЛАДНІШІ СТРУКТУРИ ДАНИХ ТА МЕТОДИ РОЗРОБКИ АЛГОРИТМІВ	
ЛР 5	Дерева. Бінарні дерева. Бінарні дерева пошуку

Індивідуальні науково-дослідні завдання (ІНДЗ)

ІНДЗ з даної навчальної дисципліни не передбачені.

Завдання для самостійної роботи студентів

Назва теми	Кількість годин
Асимптотична складність алгоритмів О-нотація, Ω , Θ , аналіз найгіршого, середнього та найкращого випадків.	6
Хеш-таблиці та методи хешування Хеш-функції, колізії, методи їх розв'язання (ланцюжки, відкрита адресація).	6
Графи та алгоритми роботи з графами Подання графів, BFS, DFS, знаходження найкоротших шляхів.	6
Купи та пріоритетні черги Бінарна купа, операції insert/extract, застосування в алгоритмах.	6
Збалансовані дерева пошуку AVL-дерева, червоно-чорні дерева, порівняння з простими BST.	6
Динамічне програмування Основні ідеї, класичні задачі (рюкзак, числа Фібоначчі, LIS).	6
Жадібні алгоритми Принцип жадібного вибору, приклади задач (коди Хаффмана, мінімальне покриття).	6
Бектрекінг (Backtracking) як метод розробки алгоритмів Принцип перебору з поверненням, дерево пошуку рішень, умови відсікання, приклади задач: задача про N ферзів, генерація перестановок і комбінацій, sudoku.	6
Рекурсія та перетворення рекурсивних алгоритмів у ітеративні Стек викликів, хвостова рекурсія, оптимізація.	6
Абстрактні типи даних (ADT) та їх реалізація Інтерфейси, інкапсуляція, порівняння різних реалізацій одного ADT.	6
Разом	60

Самостійна робота студента полягає в опрацюванні лекційного матеріалу, більш детальному розгляді окремих питань курсу, підготовці до виконання та захисту лабораторних робіт.

Самостійна робота студента загалом складає 60 годин. Розподіл цих годин за видами робіт: опрацювання лекційного матеріалу, більш детальний розгляд окремих питань курсу – 30 годин; підготовка до виконання та захисту лабораторних робіт – 30 годин.

Оцінювання самостійної роботи студента є складовою частиною оцінювання захисту лабораторних робіт (30% оцінки кожної лабораторної роботи складає оцінювання самостійної роботи).

МЕТОДИ НАВЧАННЯ

Методи, що використовуються у освітньому процесі:

- методи формування професійної компетентності – пояснення, демонстрація, візуалізація;
- методи формування практичних умінь та навичок – розв’язування задач, виконання практичних завдань.

СИСТЕМА КОНТРОЛЮ ТА ОЦІНЮВАННЯ

Методами та формами поточного контролю є:

- усні поточні опитування;
- захист виконаних лабораторних робіт.

Формою підсумкового контролю є усний залік.

ПОЛІТИКА ОСВІТНЬОГО ПРОЦЕСУ

Політика щодо відвідування. Відвідування занять є обов’язковим. За наявності поважних причин (хвороба, індивідуальний графік навчання) навчання, за погодженням із викладачем, може відбуватись у змішаній (очно-дистанційній) формі. За необхідності, з метою з’ясування всіх незрозумілих під час самостійної роботи питань, потрібно відвідувати консультації викладача.

Політика щодо академічної доброчесності. Студент зобов’язаний самостійно, своєчасно та добросовісно виконувати усі отримані завдання, дотримуючись принципів академічної доброчесності. Під час проведення контрольних заходів заборонені використання сторонньої допомоги, навчальної літератури, конспектів, мобільних пристроїв, інших джерел інформації.

Політика щодо дедлайнів та перескладання. Якщо студент був відсутній на заняттях (з будь-якої причини), він повинен самостійно опрацювати пропущений матеріал та прозвітувати викладачу про виконання відповідних завдань у встановлені ним терміни під час консультацій. Якщо студент з поважної причини пропустив захист лабораторної роботи, він може з дозволу викладача захистити її під час консультації. Відсутність студента на захисті лабораторної роботи без поважної причини відповідає оцінці «0 балів», тобто призводить до заборгованості, яка повинна бути ліквідована студентом до початку підсумкового контролю з наступного модуля. Кінцевий термін ліквідації заборгованості з модульного контролю обмежується початком заліково-екзаменаційної сесії. Лабораторні роботи, які захищаються невчасно без поважних причин, оцінюються на нижчу оцінку (від -10% до -50% від максимальної кількості балів –

залежно від терміну затримки здачі). Перезахист лабораторних робіт не дозволяється. Складання (перескладання) заліку відбувається за встановленим деканатом розкладом.

КРИТЕРІЇ ОЦІНЮВАННЯ ПОТОЧНОГО ТА ПІДСУМКОВОГО КОНТРОЛЮ

Оцінювання знань студентів здійснюється на основі результатів поточного, модульного та підсумкового контролю. Оцінювання здійснюється за програмним матеріалом навчальної дисципліни, засвоєння якого перевіряється пропонованими видами контролю.

Поточний контроль здійснюється під час проведення лабораторних занять і перевірки самостійної роботи студентів, а також під час читання лекцій. Модульний контроль здійснюється за результатами виконаних модульних контрольних робіт та перевірки лабораторних робіт і проєктів. Завданнями поточного та модульного контролю є перевірка рівня розуміння та засвоєння лекційного матеріалу, набуття практичних навичок і досвіду виконання індивідуальних і комплексних задач.

Завданням підсумкового контролю (іспиту) є перевірка розуміння студентом програмного матеріалу в цілому, здатності успішно розв'язувати поставлені практичні задачі та комплексно використовувати отримані знання.

Оцінювання знань здійснюється за 100-бальною шкалою. Результати роботи впродовж навчального семестру оцінюються в ході поточного та модульного контролю на інтервалі оцінок від 0 до 60 балів, а результати підсумкового контролю (іспиту) оцінюються максимум у 40 балів. Загальна підсумкова оцінка з навчальної дисципліни виставляється за загальною сумою балів поточного та модульного контролю і підсумкового контролю.

Розподіл балів з навчальної дисципліни

Поточне тестування та самостійна робота						Іспит	Сума
Змістовий модуль №1		Змістовий модуль №2		Змістовий модуль №3			
ЛР 1	ЛР 2	ЛР 3	ЛР 4	ЛР 5	КР	40	100
10	10	10	5	5	20		

Іспит (40 балів) складається з онлайн-тестування на 20 балів та комплексного практичного завдання на 20 балів. Оцінювання результатів проходження онлайн тестування здійснюється автоматично, на основі кількості наданих правильних відповідей на питання тесту. За суттєві помилки в розв'язуванні практичного завдання знімається 10-14 балів. За несуттєві помилки в розв'язуванні практичного завдання знімається 2-8 балів. Часткове розв'язання практичного завдання із грубими помилками оцінюється не вище 6 балів.

Загальна підсумкова оцінка з навчальної дисципліни виставляється за загальною сумою балів, набраних студентом, згідно з наступною таблицею:

Шкала оцінювання: національна та ЄКТС

Оцінка за національною шкалою	Оцінка за шкалою ECTS	
	Оцінка (бали)	Пояснення за розширеною шкалою
відмінно	A (90-100)	відмінно
добре	B (80-89)	дуже добре
	C (70-79)	добре
задовільно	D (60-69)	задовільно
	E (50-59)	достатньо
незадовільно з можливістю повторного складання	FX (35-49)	(незадовільно) з можливістю повторного складання
незадовільно з обов'язковим самостійним опрацюванням освітнього компонента до перескладання	F (0-34)	(незадовільно) з обов'язковим самостійним опрацюванням освітнього компонента до перескладання

ПЕРЕЛІК ПИТАНЬ ДЛЯ САМОКОНТРОЛЮ ТА ПІДСУМКОВОГО КОНТРОЛЮ НАВЧАЛЬНИХ ДОСЯГНЕНЬ СТУДЕНТІВ

1. Дати визначення однозв'язного списку та описати його основні операції.
2. Пояснити відмінності між однозв'язним і двозв'язним списками.
3. Описати принцип роботи ітераторів у списках та їх переваги.
4. Навести алгоритми вставки та видалення елементів у списках з ітераторами.
5. Дати визначення стеку та описати його основні операції.
6. Пояснити принцип роботи черги та її відмінність від стеку.
7. Пояснити поняття оберненої польської нотації та алгоритм обчислення виразів у ній.
8. Дати визначення кільцевого списку та описати сфери його застосування.
9. Пояснити особливості реалізації кільцевих списків з ітераторами.
10. Навести класифікацію алгоритмів сортування та охарактеризувати їх основні властивості.
11. Пояснити принцип роботи методу декомпозиції «розділяй та володарюй».
12. Навести приклади алгоритмів, побудованих за методом «розділяй та володарюй».
13. Дати визначення дерева як структури даних та описати його основні характеристики.
14. Пояснити будову бінарного дерева та основні способи його обходу.
15. Дати визначення бінарного дерева пошуку та пояснити принципи вставки, пошуку й видалення елементів.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Краснокутська І.В. Алгоритми та структури даних // (лабораторний практикум). – Чернівці: Чернівецький національний ун-т, 2017. – 48 с.
2. Крєневич А.П. Алгоритми і структури даних. Підручник. – К.: ВПЦ "Київський Університет", 2021. – 200 с.
3. Кормен Т.Г., Лейзерсон Ч.Е., Рівест Р.Л., Стайн К. Вступ до алгоритмів. – К. : К. І. С., 2019. – 1288 с.
4. Karumanchi N. Data Structures and Algorithms Made Easy – CareerMonk Publications, 2016. – 418 p.
5. Wengrow J. Common-Sense Guide to Data Structures and Algorithms. – Pragmatic Bookshelf, 2020. – 481 p.
6. Drozdek A. Data Structures and Algorithms in C++. – Cengage Learning, 2005. – 758 p.
7. Levitin A. Introduction to the Design and Analysis of Algorithms. – Pearson, 2012. – 565 p.
8. Sedgewick R., Wayne K. Algorithms. – Addison-Wesley, 2014. – 955 p.
9. Aho, A.V., Hopcroft J.E., Ullman J.D. Data Structures and Algorithms. – AddisonWesley, Reading, Mass, 2007. – 427 p.
10. Кормен Т.Х., Лейзерсон Ч.І., Рівест Р.Л., Штайн К. Алгоритми: побудова та аналіз. – 3-е вид. – К.: Вид. дім "Києво-Могилянська академія", 2021. – 1328 с.
11. Пасічник В.В. Основи програмування та структури даних на C++. – Львів: Видавництво Львівської політехніки, 2015. – 412 с.
12. Кудрявцев В.М. **Програмування** та алгоритмічні структури даних. – К.: Либідь, 2020. – 560 с.
13. Воронович О.І. Структури даних та алгоритми в C++. – К.: Видавничий дім "СофтПрес", 2016. – 448 с.

Додаткова (допоміжна)

1. Weiss M.A. Data Structures and Algorithm Analysis in C++. – 4th ed. – Boston: Addison-Wesley, 2013. – 656 p.
2. Skiena S.S. The Algorithm Design Manual. – 2nd ed. – London: Springer, 2008. – 730 p.
3. Коротєєва Т.О. Алгоритми та структури даних: навч. посібник. – Львів: Львівської політехніки, 2014. – 280 с.
4. Preiss B.R. Data Structures and Algorithms with Object-Oriented Design Patterns in Python. – 2003. – 566 с.
5. Problem Solving with Algorithms and Data Structures using Python [Електронний ресурс] – Режим доступу до ресурсу:
<https://runestone.academy/runestone/books/published/pythonds/index.html>.
6. Вихідний код програм [Електронний ресурс] – Режим доступу до ресурсу:
<https://github.com/krenevych/algo>.
7. Lee K.D. Data Structures and Algorithms with Python. – Springer, 2015. – 363 с.

ІНФОРМАЦІЙНІ РЕСУРСИ

1. Електронний курс в системі Moodle
<https://moodle.chnu.edu.ua/course/view.php?id=7301>
2. Візуалізація алгоритмів та структур даних Національного університету Сінгапуру VisuAlgo
<https://visualgo.net/en>

ПОЛІТИКА АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ

Дотримання політики щодо академічної доброчесності учасниками освітнього процесу при вивченні освітнього компонента регламентовано чинними документами ЧНУ:

- «Кодекс академічної доброчесності та етики Чернівецького національного університету імені Юрія Федьковича»
<https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/kodeks-akademichnoi-dobrochesnosti-ta-etyky-chernivetskoho-natsionalnoho-universytetu-imeni-yurii-fedkovycha/>
- «Положенням про виявлення та запобігання академічного плагиату у Чернівецькому національному університеті імені Юрія Федьковича»
<https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polozhennia-pro-vyivlennia-ta-zapobihannia-akademichnomu-plahiatu-u-chernivetskomu-natsionalnomu-universyteti-imeni-yurii-fedkovycha/>
- «Політика використання штучного інтелекту в Чернівецькому національному університеті імені Юрія Федьковича»
<https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polityka-vykorystannia-shtuchnoho-intelektu-v-chernivetskomu-natsionalnomu-universyteti-imeni-yurii-fedkovycha-zi-zminamy-zatverdzheny-vchenoiu-radoiu-chnu-imeni-yurii-fedkovycha/>

Використання здобувачами освіти інструментів штучного інтелекту (ШІ)

під час виконання навчальних завдань:

- допускається за умови дотримання принципів академічної доброчесності (здобувач освіти несе відповідальність за зміст, достовірність і результати поданої роботи, а у разі використання ШІ зобов'язаний зазначити факт і спосіб його застосування);
- не допускається використання ШІ як заміни самостійного виконання завдань або з метою порушення вимог академічної доброчесності;
- додаткові обмеження чи вимоги щодо використання ШІ можуть визначатися викладачем в освітньому процесі для конкретних видів навчальної діяльності відповідно до особливостей дисципліни.