

Чернівецький національний університет імені Юрія Федьковича
Факультет математики та інформатики
Кафедра прикладної математики та інформаційних технологій

“ЗАТВЕРДЖУЮ”

**Декан факультету
математики та інформатики**

_____ (Мартинюк О.В.)
(підпис) (прізвище, ініціали)

“ ” _____ 2026 р.

РОБОЧА ПРОГРАМА
навчальної дисципліни

Створення власної мови програмування

(назва навчальної дисципліни)

Вибіркова навчальна дисципліна

(вказати: обов’язкова / вибіркова)

Освітньо-професійна програма Технології програмування та комп’ютерне моделювання

(назва програми)

Спеціальність 113 – Прикладна математика

(вказати: код, назва)

Галузь знань 11 – Математика та статистика

(вказати: шифр, назва)

Рівень вищої освіти перший (бакалаврський)

(вказати: перший бакалаврський/другий магістерський)

Факультет математики та інформатики

(назва факультету, на якому здійснюється підготовка фахівців за вказаною освітньо-професійною програмою)

Мова навчання українська

(вказати: на якій мові читається дисципліна)

Чернівці 2026

Робоча програма із вибіркової навчальної дисципліни
«Створення власної мови програмування» 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", затвердженої Вченою радою Чернівецького
національного університету імені Юрія Федьковича 28 квітня 2025 р.,
протокол №5.

Розробник: Тетяна Миколаївна Сопронюк доцент, канд. – фіз. мат. наук, доцент
(П.І.Б. авторів, посада, науковий ступінь, вчене звання)

Викладач, що забезпечує читання даної навчальної дисципліни:

Тетяна Миколаївна Сопронюк , доцент, канд. – фіз. мат. наук, доцент
(П.І.Б. авторів, посада, науковий ступінь, вчене звання)

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
факультету математики та інформатики _____ Тоня ФРАТАВЧАН

Затверджено Вченою радою факультету математики та інформатики
Протокол № від “__” _____ 2026 року

Голова Вченої ради
факультету математики та інформатики _____ Ольга МАРТИНЮК

Мета навчальної дисципліни: Мета - опанувати основні принципи побудови компіляторів, елементи теорії формальних мов та теорію скінченних автоматів, навчитися будувати спрощені лексичні і синтаксичні аналізаторів, продемонструвати застосування основних принципів побудови компіляторів на прикладі розробки інтерпретатора простої мови програмування SPL (інтерпретатор має обмежені можливості, і це дозволяє охопити весь процес його побудови).

Пререквізити: системне програмування (теорія скінченних автоматів, формальні граматики, регулярні вирази).

Результати навчання (*формулювання результатів навчання у вигляді переліку загальних та фахових компетентностей, програмних результатів відповідно до ОПП*).

У результаті вивчення навчальної дисципліни студент повинен

знати: основні фази компіляції, принципи побудови лексичних і синтаксичних аналізаторів та компіляторів.

вміти: застосовувати теорію формальних мов до вирішення задач аналізу, програмувати алгоритми перетворення скінченних автоматів, праволінійних граматики та регулярних виразів; розробляти власні найпростіші аналізатори та інтерпретатори.

Студенти повинні оволодіти програмним матеріалом, застосувати вивчені алгоритми до модельних прикладів, запрограмувати частину алгоритмів, виконати залікову тестову контрольну роботу.

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

загальні компетентності:

- ЗК01. Здатність учитися і оволодівати сучасними знаннями.
- ЗК02. Здатність застосовувати знання у практичних ситуаціях.
- ЗК03. Здатність генерувати нові ідеї (креативність).
- ЗК06. Здатність до абстрактного мислення, аналізу та синтезу.
- ЗК10. Навички у використанні інформаційних і комунікаційних технологій.

фахові компетентності:

- ФК01. Здатність використовувати й адаптувати математичні теорії, методи та прийоми для доведення математичних тверджень і теорем.
- 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.
- ФК07. Здатність експлуатувати та обслуговувати програмне забезпечення автоматизованих та інформаційних систем різного призначення.
- 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.
- ФК17. Здатність до використання новітніх інформаційно-комунікаційних технологій.

та отримуються наступні **програмні результати навчання:**

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

ПРН14. Виявляти здатність до самонавчання та продовження професійного розвитку.

ПРН16. Демонструвати навички взаємодії з іншими людьми, уміння працювати в команді.

Опис навчальної дисципліни Загальна інформація

Назва навчальної дисципліни _____												
Форма навчання	Рік підготовки	Семестр	Кількість			Кількість годин						Вид підсумкового контролю
			кредитів	годин	змістових	лекції	практичні	семінарськ	лаборатор	самостійна робота	індивідуал	
Денна	4	7	3	90	2	30	-	-	16	44	-	Залік

Структура змісту навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	денна форма					
	усього	у тому числі				
		л	п	ла б	інд	с.р.
1	2	3	4	5	6	7
Модуль 1						
Змістовий модуль 1. Лексичний і синтаксичний аналіз						
Тема 1. Лексичний аналіз і скінченні автомати. Поняття лексичного аналізатора. Побудова лексичних аналізаторів, що розпізнають лексему одного типу. Побудова лексичного аналізатора, що розпізнає декілька типів лексем у тексті. Програмна реалізація детермінованого скінченного автомата з декількома типами заключних станів. Скінченні автомати для розпізнавання ключових слів у тексті.	12	4	-	4	-	4
Тема 2. Синтаксичний аналіз і контекстно-вільні граматики. Розбір та інтерпретація математичних формул. Граматики Хомського і LA(1)-граматики. Синтаксичний аналізатор	12	4	-	4	-	4

розпізнавання виразів. Середовище розробки аналізаторів ANTLRWorks.						
Тема 3. Побудова інтерпретатора математичних формул. Побудова спрощеного синтаксичного аналізатора для виконання числових обчислень і обчислення значення функції в точці. Лексичний аналіз формули. Синтаксичний аналіз формули методом рекурсивного спуску.	8	2	-	-	-	6
Тема 4. Обчислення регулярних виразів над формалізмами автоматних мов. Створення ієрархії класів для формалізмів автоматних мов. Робота з об'єктами класів “Регулярний вираз”, “Праволінійна граматика”, “Недетермінований скінчений автомат”. Обчислення глобального регулярного виразу методом рекурсивного спуску	8	2	-	-	-	6
Разом за змістовим модулем 1	40	12	-	8	-	20
Змістовий модуль 2. Створення інтерпретатора мови SPL						
Тема 1. Можливості інтерпретатора мови програмування SPL. Неформальний опис мови SPL . Приклади SPL-програм. Склад інтерпретатора	10	4	-	2	-	4
Тема 2. Фаза лексичного аналізу. Типи лексем та їх кодування. Таблиця ідентифікаторів. Приклад лексичного аналізу SPL-програми	8	2	-	1	-	5
Тема 3. Синтаксичний аналіз мови SPL. Формальний синтаксис мови SPL. Аналіз об'яв. Аналіз операторів і виразів. Приклад граматичного розбору одного оператора мови SPL. Застосування середовища розробки аналізаторів ANTLRWorks для опису граматики мови SPL.	10	4	-	2	-	4
Тема 4. Інтерпретація. SPL-процесор. Виконання SPL-програми. Створення SPL-програми. Приклад проміжного коду для SPL програми. Розробка компілятора SPL для платформи .NET	12	4	-	2	-	6
Тема 5. Генерація проміжного коду. Робота з таблицями. Розширення аналізу описів. Розширення аналізу операторів та виразів.	10	4	-	1	-	5
Разом за змістовим модулем 2	50	16	-	8	-	26
<i>Усього годин</i>	90	30	-	16	-	44

Теми лекційних занять

Тема 1.1. Лексичний аналіз і скінченні автомати. Поняття лексичного аналізатора. Побудова лексичних аналізаторів, що розпізнають лексему одного типу. Побудова лексичного аналізатора, що розпізнає декілька типів лексем у тексті. Програмна реалізація

детермінованого скінченного автомата з декількома типами заключних станів. Скінченні автомати для розпізнавання ключових слів у тексті.

Тема 1.2. Синтаксичний аналіз і контекстно-вільні граматики. Розбір та інтерпретація математичних формул. Граматики Хомського і LA(1)-граматики. Синтаксичний аналізатор розпізнавання виразів. Середовище розробки аналізаторів ANTLRWorks.

Тема 1.3. Побудова інтерпретатора математичних формул. Побудова спрощеного синтаксичного аналізатора для виконання числових обчислень і обчислення значення функції в точці. Лексичний аналіз формули. Синтаксичний аналіз формули методом рекурсивного спуску.

Тема 1.4. Обчислення регулярних виразів над формалізмами автоматних мов. Створення ієрархії класів для формалізмів автоматних мов. Робота з об'єктами класів "Регулярний вираз", "Праволінійна граMATика", "Недетермінований скінчений автомат". Обчислення глобального регулярного виразу методом рекурсивного спуску

Тема 2.1. Можливості інтерпретатора мови програмування SPL. Неформальний опис мови SPL . Приклади SPL-програм. Склад інтерпретатора

Тема 2.2. Фаза лексичного аналізу. Типи лексем та їх кодування. Таблиця ідентифікаторів. Приклад лексичного аналізу SPL-програми

Тема 2.3. Синтаксичний аналіз мови SPL. Формальний синтаксис мови SPL. Аналіз об'яв. Аналіз операторів і виразів. Приклад граматичного розбору одного оператора мови SPL. Застосування середовища розробки аналізаторів ANTLRWorks для опису граматики мови SPL.

Тема 2.4. Інтерпретація. SPL-процесор. Виконання SPL-програми. Створення SPL-програми. Приклад проміжного коду для SPL програми. Розробка компілятора SPL для платформи .NET

Тема 2.5. Генерація проміжного коду. Робота з таблицями. Розширення аналізу описів. Розширення аналізу операторів та виразів.

Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Графічне середовище розробки граматик ANTLRWorks	2
2	Побудова спрощеного лексичного аналізатора	4
3	Побудова спрощеного синтаксичного аналізатора	4
4	Елементи компіляції	4

Самостійна робота

№ з/п	Назва алгоритмів для самостійного вивчення (п.1) і завдань для самостійного виконання (п.2)	Кількість годин
1	1. Побудова множин FIRST і FOLLOW 2. Обчислення FOLLOW 3. Побудова праволінійної граматики за скінченим автоматом	22

	<u>4. Видалення лівої рекурсії</u> <u>5. Ліва факторизація</u> <u>6. Алгоритми приведення граматики до нормальної форми Хомського</u> <u>6.1. Алгоритм видалення безпосередніх лівих рекурсій за допомогою е-правил.</u> <u>6.2. Алгоритм видалення безпосередніх лівих рекурсій за допомогою розширених правил.</u> <u>6.3. Алгоритм видалення безкорисних символів і правил</u> <u>6.4. Алгоритм видалення недосяжних символів</u> <u>6.5. Алгоритм видалення е-правил</u> <u>6.6. Алгоритм усунення ланцюгових правил</u> <u>7. Нормальна форма Хомського</u> <u>8. Алгоритм перетворення до нормальної форми Хомського</u> <u>9. Опис програмної реалізації алгоритмів</u> <u>9.1. Програмна реалізація алгоритму видалення лівої рекурсії за допомогою е-правил</u> <u>9.2. Програмна реалізація алгоритму видалення лівої рекурсії за допомогою розширених правил</u>	
2	1. Розширення можливостей спрощеного синтаксичного аналізатора для виконання символічних обчислень з використанням стандартних функцій [3]. 2. Генерація проміжного коду для SPL-програми. Робота з таблицями. Розширення аналізу описів. Розширення аналізу операторів та виразів. Виконання коду [3]. 3. Аналіз проміжних файлів роботи SPL-інтерпретатора, які утворюються для конкретної програми з лабораторної роботи №6 [3]. 4. Автоматне програмування. Конструкції while-switch-case. Перетворення ітераційних алгоритмів в автоматні [34]. 5. Виконання домашніх завдань, які регулярно задаються під час викладання лекцій, для закріплення матеріалу і засвоєння наведених алгоритмів [3]. 6. Оформлення лабораторних робіт. 7. Знайомство з матеріалами магістерських, дипломних і курсових робіт, виконаних на кафедрі ПМ по тематиці “Системне програмування” [11]. 8. Робота з посібниками [2,3] і додатковими матеріалами, які винесені у локальну мережу за адресою <i>Сетевое окружение \ Программная группа Аpm\ Math_02\ Zavr_Lab\ SystProg</i>, на сайт “Системне програмування” і на сайт дистанційної освіти	22
	Разом	44

Критерії оцінювання результатів навчання з навчальної дисципліни

Підсумкова оцінювання — це тестове завдання.

Тести відкриваються екзаменатором і оцінюються системою Moodle.

Посилання на тест: (<http://e-learning.fpm.chnu.edu.ua/mod/quiz/view.php?id=15692>).

Оцінювання відбувається автоматично. Правильна відповідь на кожне питання оцінюється у 1 бал. Тести стосуються лекційного матеріалу та завдань, які давалися на протязі семестру. Послідовність питань вибирається випадковим чином із підготовленого банку питань.

Захист лабораторної/практичної роботи та критерії оцінювання

- Здача лабораторної роботи та практичного заняття завдання проводиться під час заняття згідно з календарним планом.
- Для захисту лабораторної роботи кожен студент має самостійно виконати лабораторну роботу і здати її викладачу **на занятті**.
- Не допускається заочне прийняття програм (електронною поштою) без запуску програм з різними вхідними даними.
- Під час здачі програми викладач зобов'язаний перевіряти здатність студента орієнтуватися у власній програмі, пропонуючи йому виконати нескладні зміни, розраховані на 5-10 хвилин поточного заняття.
- При необхідності виконання частини завдання або усього завдання у **робочому** зошиті, бали виставляти у зошиті, вказуючи число і підпис.
- Під час здачі лабораторної роботи студент повинен
 - вміти пояснити постановку задач, які розв'язувались в лабораторній роботі; алгоритм розв'язування задач; програмну реалізацію завдання;
 - продемонструвати розуміння програми та обґрунтувати зроблені висновки;
 - відповісти на питання, які належать до виконання лабораторної роботи та додаткові теоретичні питання, якщо розданий перелік таких питань.
- Якщо студент не розуміє алгоритму розв'язання задачі, не орієнтується в програмній реалізації, але є у наявності правильно виконувана програма, то робота зараховується не більше як на 30%.
- Якщо студент розуміє задачу і алгоритм її виконання, але не орієнтується (слабо орієнтується) в практичній частині (програмній реалізації), то оцінка знижується до 50%.
- Якщо програма не працює, або працює частково і студент може пояснити алгоритм, роботу оцінювати частково, в залежності від об'єму і якості коду.

- Кількість балів за лабораторну роботу визначає викладач в процесі здачі. Оцінка повідомляється студенту.
- За невчасний захист лабораторних робіт у межах модуля допускається знімати по одному балу за кожне прострочене заняття, якщо робота оцінюється до 10 балів, і по 1,5-2 бали, якщо робота оцінюється в межах від 11 до 20 балів, але не більше половини балів.

Шкала оцінювання: національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка ЄКТС	Оцінка за національною шкалою
		для заліку
90 – 100	A	зараховано
80-89	B	
70-79	C	
60-69	D	
50-59	E	
35-49	FX	не зараховано з можливістю повторного складання
0-34	F	не зараховано з обов'язковим самостійним опрацюванням освітнього компоненту до перескладання

Засоби оцінювання

Під час проведення лекцій використовуються пасивний та активний методи навчання. Консультаційна робота. Під час виконання студентами лабораторних робіт використовується активні методи навчання. Проведення модульних лабораторних та практичних робіт та навчальна робота під час їх прийому.

Форми поточного та підсумкового контролю

1. . Оцінювання на лабораторних заняттях в обох модулях.
2. Тестові завдання засобами системи Moodle.

Розподіл балів, які отримують студенти

Поточне оцінювання та самостійна робота									Залік	Сума
Змістовий модуль №1				Змістовий модуль № 2					Тести	
T1	T2	T3	T4	T1	T2	T3	T4	T5		
10	10	10	10	10	10	10	5	5	20	100

Рекомендована література

Базова (основна)

1. Сопронюк Т.М. Мовні процесори та формальні мови: від теорії до практики : навч. посібник. Чернівці : Чернівець. нац. ун-т ім. Ю. Федьковича, 2025. 198 с. ISBN 978-966-423-942-1 <https://archer.chnu.edu.ua/xmlui/handle/123456789/12079>
2. Сопронюк Т.М. Елементи теорії компіляції: Навчальний посібник. – Чернівці: ЧНУ, 2008. – 84 с.
3. Сопронюк Т.М. Елементи теорії компіляції: Навчальний посібник. – Чернівці: ЧНУ, 2008. – 84 с.
4. A.V. Aho and J. D. Ullman. The Theory of Parsing, Translation, and Compiling. Vol. 1. Englewood Cliffs, N.J.: Prentice Hall, 1972. - 460 p. <https://www.amazon.com/Theory-Parsing-Translation-Compiling/dp/0139145567>
4. P. J. Denning, J. B. Dennis, and J. E. Qualitz. Machines, Languages, and Computation. Englewood Cliffs, N.J.: Prentice Hall. 3. M., 1978.
5. M. A. Harrison. Introduction to Formal Language Theory. Reading, Mass.: Addison-Wesley, 1978.
6. W. Homenda, W. Pedrycz. Automata theory and formal languages, De Gruyter, 2022.
7. J. E. Hopcroft and J. D. Ullman. Introduction to Automata Theory, Languages and Computation. Reading, Mass.: Addison-Wesley, 1979.
8. R. Hunter. The Design and Construction of Compilers. Chichester, New York: John Wiley, 1981.
9. Z. Kohavi and N. K. Jha. Switching and Finite Automata Theory. Third Edition. New York: Cambridge University Press, 2010.
10. P. Linz. An Introduction to Formal Languages and Automata-Peter Linz Univ. of California at Davis. Jones & Bartlett Learning, 2016.
- A. Pettorossi. Automata theory and formal languages: fundamental notions, theorems, and techniques. Springer, 2022.
11. G. E. Revesz. Introduction to Formal Languages. New York: McGrawHill, 1983.
12. Теорія цифрових автоматів та формальних мов. Вступний курс: навчальний посібник С.Ю Гавриленко, А.М. Клименко, Н.Ю. Любченко та ін. – Харків: НТУ ХПІ, 2007. – 176 с.
13. Олег Гутік, Формальні мови та автомати. Ел. Посібник. Львів, 2022.
14. Олег Гутік, Коди та автомати: основи алгебраїчної теорії. Ел. Посібник. Львів, 2021.
15. Сопронюк Т. М., Сопронюк А. Ю., Дробот А. В. Фази побудови мовного процесора для платформи .NET// Буковинський матем. журнал. — 2023. — Т.11, No2. — С. 71–84.
16. Формальні мови, граматики та автомати. [Електронний ресурс]. – Режим доступу: <https://repository.kpi.kharkov.ua/server/api/core/bitstreams/5847efdd-6ff5-4f7f-9de8-6f6555ad4cc0/content>
17. Online compiler and debugger [Електронний ресурс] Режим доступу: <https://www.onlinegdb.com>
18. Вікіпедія. Формальні граматики. Доступно за посиланням: http://uk.wikipedia.org/wiki/Формальні_граматики. [Дата звернення: 20 січня 2025].
19. Вікіпедія. Скінченний автомат. Доступно за посиланням: https://uk.wikipedia.org/wiki/Скінченний_автомат. [Дата звернення: 20 січня 2025].
20. Сопронюк Т. М., Сопронюк А. Ю., Дробот А. В. Фази побудови мовного процесора для платформи .NET// Буковинський матем. журнал. — 2023. — Т.11, No2. — С. 71–84.[Електронна версія: DOI: <https://doi.org/10.31861/bmj2023.02.07>]

Інформаційні ресурси

17. <http://www-db.stanford.edu/~ullman/ialc.html> – Introduction to Automata Theory, Languages, and Computation. Slides and Lecture Notes. (Stanford University)
18. <https://www.pdfdrive.com/the-theory-of-parsing-translation-and-compiling-volume-1-parsing-e157156882.html> - The Theory of Parsing, Translation, and Compiling (Volume 1): Parsing by Alfred V. Aho & Jeffrey D. Ullman. 1973– 562 p.
19. <http://www.unicyb.kiev.ua/Library/PROG/Zmist.htm> – А.Б.Ставровський. Посібник з програмування (факультет кібернетики Київського національного університету)
20. http://uk.wikipedia.org/wiki/Формальні_граматики
21. <http://courses.cs.vt.edu/~cs4114/lectures/> – Formal Languages and Automata Theory Course Lecture notes
22. http://khpi-iip.mipk.kharkiv.edu/library/datastr/book_sod/kgsu/oglav.html – Мова програмування C++. Динамічні структури даних

Додатково

Методичне забезпечення

1. Сопронюк Т.М. Мовні процесори та формальні мови: від теорії до практики : навч. посібник. Чернівці : Чернівець. нац. ун-т ім. Ю. Федьковича, 2025. 198 с. ISBN 978-966-423-942-1 <https://archer.chnu.edu.ua/xmlui/handle/123456789/12079>
2. Сопронюк Т.М. Елементи теорії компіляції: Навчальний посібник. – Чернівці: ЧНУ, 2008. – 84 с.
3. Сопронюк Т.М. Елементи теорії формальних мов: Навчальний посібник. – Чернівці: ЧНУ, 2008. – 84 с.
4. Сопронюк Т. М., Сопронюк А. Ю., Дробот А. В. Фази побудови мовного процесора для платформи .NET// Буковинський матем. журнал. — 2023. — Т.11, No2. — С. 71–84.
5. Тестові завдання (Система Moodle)
6. Презентації лекцій (Система Moodle)
7. Відео-лекції на Google диску
8. Сертифікат про закінчення курсів професора Стенфордського університету Дж.Ульмана “Automata” (<https://drive.google.com/file/d/0B-cxrXmP0J7XRUI2YlNGem1wb3c/view?resourcekey=0--NXsRlO20hZHdxuVKIDQ4Q>)

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/universitytet/normatyvni-dokumenty/kodeks->

[akademichnoi-dobrochesnosti-ta-etyky-chernivetskoho-natsionalnoho-universytetu-imeni-yurii-fedkovycha/](https://www.chnu.edu.ua/universytet/normatyvni-dokumenty/polozhennia-pro-vyivlennia-ta-zapobihannia-akademichnomu-plahiatu-u-chernivetskomu-natsionalnomu-universyteti-imeni-yurii-fedkovycha/)

- «Положення про виявлення та запобігання академічного плагіату 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-shtuchoho-intelektu-v-chernivetskomu-natsionalnomu-universyteti-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.