

Чернівецький національний університет імені Юрія Федьковича

(повне найменування вищого навчального закладу)

Факультет математики та інформатики

(назва інституту / факультету)

Кафедра прикладної математики та інформаційних технологій
(назва кафедри)

“ЗАТВЕРДЖУЮ”

Деканка факультету математики та інформатики

Мартинюк О.В.

“ ” 2026 року

РОБОЧА ПРОГРАМА

навчальної дисципліни

Об’єктно-орієнтоване програмування

(назва навчальної дисципліни)

обов’язкова

(вказати: обов’язкова / вибіркова)

Освітньо-професійна програма Технології програмування та комп’ютерне моделювання

(назва програми)

Спеціальність F1– Прикладна математика

(вказати: код, назва)

Галузь знань F – Інформаційні технології

(вказати: шифр, назва)

Рівень вищої освіти перший бакалаврський

(вказати: перший бакалаврський/другий магістерський)

факультет математики та інформатики

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

Мова навчання українська

(вказати: на якій мові читається дисципліна)

Чернівці 2026 рік

Робоча програма навчальної дисципліни **Об'єктно-орієнтоване 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 No5.

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 “___” _____ червня 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: поглиблене оволодіння сучасними технологіями програмування на C++ з використанням класів, успадкування і поліморфізму. Для досягнення мети студенти повинні оволодіти програмним матеріалом, написати програми, виконати модульні контрольні роботи, здати екзамен або залік.

Курс присвячено вивченню сучасних технологій програмування на C++ (парадигми процедурного, модульного і об'єктно-орієнтованого програмування) з використанням класів, успадкування і поліморфізму. У курсі демонструється застосування об'єктно-орієнтованого підходу для розробки програмного забезпечення.

Пререквізити: програмування.

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: поняття базового і породженого класів, ієрархії класів; методику програмування, що концентрує основну увагу на зв'язках між об'єктами, а не на деталях їхньої реалізації (інкапсуляція дає можливість об'єднання даних і алгоритмів їх обробки, в результаті чого і дані, і процедури багато в чому втрачають самостійне значення);

вміти: застосовувати теоретичні знання для створення власних класів, перезавантажувати оператори для структур і класів, перевизначати функції, застосовувати шаблони функцій і класів, успадковувати класи, побудувати ієрархію класів, використовувати віртуальні функції і абстрактні класи. При написанні лабораторних робіт і, взагалі, програмних систем студент повинен вміти застосовувати об'єктно-орієнтований підхід.

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).
- ЗК04. Здатність бути критичним і самокритичним.
- ЗК05. Здатність проведення досліджень на відповідному рівні.
- ЗК07. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
- ЗК08. Знання та розуміння предметної області та розуміння професійної діяльності.
- ЗК10. Навички у використанні інформаційних і комунікаційних технологій.
- ЗК12. Визначеність і наполегливість щодо поставлених завдань і взятих обов'язків.

фахові компетентності:

- ФК02. Здатність виконувати завдання, сформульовані у математичній формі.
- ФК03. Здатність обирати та застосовувати математичні методи для розв'язання прикладних задач, моделювання, аналізу, проектування, керування, прогнозування, прийняття рішень.
- 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.
- 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.

ФК13. Здатність зрозуміти постановку завдання, сформульовану мовою певної предметної галузі, здійснювати пошук та збір необхідних вихідних даних.

ФК14. Здатність сформулювати математичну постановку задачі, спираючись на постановку мовою предметної галузі, та обирати метод її розв'язання, що забезпечує потрібні точність і надійність результату.

ФК17. Здатність до використання новітніх інформаційно-комунікаційних технологій.

ФК19. Здатність застосовувати сучасні інформаційні технології для комп'ютерної реалізації математичних моделей.

та отримуються наступні **програмні результати навчання**:

ПРН01. Демонструвати знання й розуміння основних концепцій, принципів, теорій прикладної математики і використовувати їх на практиці.

ПРН03. Формалізувати задачі, сформульовані мовою певної предметної галузі; формувати їх математичну постановку та обирати раціональний метод вирішення; розв'язувати отримані задачі аналітичними та чисельними методами, оцінювати точність та достовірність отриманих результатів.

ПРН14. Виявляти здатність до самонавчання та продовження професійного розвитку.

ПРН15. Уміти організувати власну діяльність та одержувати результат у рамках обмеженого часу

Опис навчальної дисципліни Загальна інформація

Назва навчальної дисципліни _____													
Форма навчання	Рік підготовки	Семестр	Кількість			Кількість годин						Вид підсумкового контролю	
			кредитів	годин	змістових	лекцій	практичні	семінарські	лабораторні	Самостійна робота	Індивідуальні завдання		
Денна	2	3	5	150	3	45	-	-	30	75	-	Екзамен	

Структура змісту навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	денна форма					
	усьог о	у тому числі				
		лек	п	лаб	інд	с.р
1	2	3	4	5	6	7
Змістовий модуль 1. Нові можливості C++, які не пов'язані з ООП						
Тема 1 Парадигма програмування. Процедурне, модульне і об'єктно орієнтоване програмування Вступ. Парадигма програмування. Процедурне програмування. Модульне програмування. Об'єктно-орієнтоване програмування. Загальна характеристика мови C++ як мови, що найбільш втілює ідеї об'єктно-орієнтованого програмування.	19	5		4		10
Тема 2. Розширені можливості мови C++, що не пов'язані з класами Прототипи функцій, аргументи по замовчуванню, посилання. Модифікатори const і volatile. Функціональний запис перетворення типів. Доступ до глобальних змінних, які закриті локальними. Функції, що підставляються. Оператори динамічного розподілу пам'яті. Перезавантаження функцій, шаблони функцій. Перезавантаження операторів для структур. Перезавантаження операцій потокового вводу і виводу для структур. Приклади перезавантаження операцій і функцій, використання шаблонів.	31	10		6		15
Разом за змістовним модулем 1	50	15		10		25
Змістовий модуль 2. Інкапсуляція. Поняття класу						
Тема 3. Класи і об'єкти. Створення і знищення об'єктів Поняття класу та його визначення. Специфікатори public, private. Доступ до відкритих членів класу. Функції доступу до захованих членів класу. Створення об'єктів. Статичні члени класу. Дружні і складові функції. Дружні класи. Ініціалізація та знищення об'єктів. Автоматичне і динамічне виділення пам'яті під об'єкти класу. Конструктори і деструктори.	22	8		4		10

Тема 4. Операції над класовими об'єктами Перевизначення операцій для класів. Бінарні і унарні операції. Пріоритети при перевизначенні операцій. Присвоєння і ініціалізації. Перезавантаження операторів виклику функції, індексування. Стандартні об'єкти потокового вводу-виводу C++. Перезавантаження операцій "<<", ">>" для класу користувача. Приклади проектування і побудови класів.	28	7		6		15
Разом за змістовним модулем 2	50	15		10		25
Змістовий модуль 3. Успадкування і поліморфізм						
Тема 5. Успадкування. Ієрархія класів. Поняття похідного та базового класів. Захищені члени класу. Співвідношення атрибутів доступу в базовому і похідному класах. Ієрархія класів. Віртуальні базові класи. Контейнеризація Конструктори і деструктори похідних класів. Перетворення вказівників на об'єкти класів. Приклади успадкування.	14	5		4		9
Тема 6. Поліморфізм. Віртуальні функції. Віртуальні функції, їх відмінність від звичайних функцій-членів. Віртуальні деструктори. Абстрактні функції і абстрактні класи. Поліморфні функції. Багаточисельне успадкування. Створення бібліотеки класів графічних фігур.	18	5		3		8
Тема 7. Бібліотеки класів і шаблонів C++ Простори імен. Огляд можливостей бібліотек STL, BOOST, MTL. Робота з рядками засобами стандартної бібліотеки та без них. Узагальнене програмування і стандартна бібліотека шаблонів STL. Приклади використання бібліотеки STL	18	6		3		8
Разом за змістовним модулем 3	50	15		10		25
Усього годин	150	45		30		75

Тематика семінарських занять

Семінарські заняття не передбачені навчальним планом.

Тематика практичних занять

Практичні заняття не передбачені навчальним планом.

Тематика лабораторних занять

№ модуля	Зміст модуля	Кількість	Кількість балів
		годин	
1	· Лабораторна робота 1. Структури. Зовнішні функції і функції-члени структури 4год.(3+1)	10	6(3+3)
	· Лабораторна робота 2. Перезавантаження операцій і функцій для структур. Використання шаблонів (2год.)		5
	· Лабораторна робота 3. Структури і класи 4год.(3+1).		8 (4+4)
	· Тести		10
2	· Лабораторна робота 4. Створення класів. Конструктори, деструктори, дружні функції і члени класу. Перезавантаження операцій для класів (6год.)	12	15(5+5+5)
	· Створення умови задачі для лабораторної роботи 5 (успадкування або агрегація (0.5год.))		2
	· Лабораторна робота 5. Проектування ієрархії класів. Успадкування. Агрегація (5год.)		11
	• Контрольна робота/задача. Створення класів (0.5год.)		10
3	· Лабораторна робота 6. Абстрактні класи (4год.)	8	11
	· Лабораторна робота 7. Створення графічного інтерфейсу до лабораторної роботи 4 , 5 або 6 (4год.)		7
	· Модуль контроль (тести)		15

Індивідуальні завдання

Наведені в системі дистанційного навчання.

Самостійна робота

№ п/п	Назва теми	Кількість годин /форми контролю
1	Виняткові ситуації	Застосування в лабораторних роботах
2	Робота з потоками	Застосування в лабораторних роботах
3	Графічні можливості	Застосування в лабораторних роботах
4	Створення віконних інтерфейсів (GUI)	Застосування в лабораторних роботах

Критерії оцінювання результатів навчання з навчальної дисципліни

Захист лабораторної роботи та критерії оцінювання

- Здача лабораторної роботи проводиться під час заняття згідно з календарними 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.

Критерії оцінювання знань на іспиті

Екзамен проводиться засобами дистанційної системи Moodle.

Складові частини оцінки:

1) тести на 25 балів:

- 10 балів (тематика першого модуля);
- 15 балів (тематика другого і третього модулів).

2) програмно згенерована випадкова контрольна задача на тему “Створення класів” на 10 балів.

У сумі з лабораторними роботами (65 балів) це загалом складатиме 100 балів.

Шкала оцінювання: національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка ЄКТС	Оцінка за національною шкалою
		для екзамену, курсового проекту (роботи), практики
90 – 100	A	відмінно
80-89	B	добре
70-79	C	
60-69	D	
50-59	E	задовільно
35-49	FX	
		незадовільно з можливістю повторного складання
0-34	F	незадовільно 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 контрольних робіт та навчальна робота під час прийому лабораторних робіт.

Форми поточного та підсумкового контролю

1. Контрольна робота.
2. Оцінювання на лабораторних заняттях в 3-х модулях.
3. Тестові завдання на заліку засобами системи Moodle.
4. Підсумкове тестування на екзамені засобами системи Moodle.

Поточне тестування та самостійна робота								
Змістовий модуль 1			Змістовий модуль 2		Змістовий модуль 3		Підсумковий тест (екзамен)	Сума
T1	T2	T3	T4	T5	T6	T7	35	100
6	5	8	15	13	11	7		

T1, T2 ... T7 – теми змістових модулів.

Рекомендована література Базова (основна)

1. Сопронюк Т.М. Об'єктно-орієнтоване програмування на C++ : навч. посібник / Т. М. Сопронюк. – Чернівці : Чернівецький національний ун-т, 2013. – 175 с. (з грифом МОНУ, лист №1/11-17600 від 18.11.2013)
2. **Sopronyuk T.** [Sopronyuk T. Object-oriented programming in C++:: Textbook](#) / Translated by Nonna Shulga: CreateSpace, 2014. – 130 p. ISBN-13: 978-1502520906 (переклад навчального посібника з

- грифом МОНУ:** Сопронюк Т.М. Об'єктно-орієнтоване програмування на C++ : навч. посібник / Т. М. Сопронюк. – Чернівці : Чернівецький національний ун-т, 2014. – 176 с.)
3. [150 C++ Programming Assignments. Variants of tasks & Examples of the code](#): Textbook / Authored by Tatyana Sopronyuk, Translated by Nonna Shulga: CreateSpace, 2015. – 73 p. ISBN-13: 978-1515254065 (CreateSpace-Assigned) ISBN-10: 1515254062.
 4. **Sopronyuk T.** [C++ Programming: Theory and Assignments](#) // Authored by Tatyana Sopronyuk, Translated by Nonna Shulga: Independently published, 2020. – 178 p. ISBN-13: 979-8651045624 ,
 5. Глинський Я.М., Анохін В.Є., Ряжська В.А. C++ і C++ Builder. -Львів: Деол, 2003.- 192 с.
 6. Ткачук В.М. Програмування на C++: Лабораторний практикум. Видавництво Прикарпатського національного університету імені Василя Стефаника, 2011.- 160с
 7. ГрицюкЮ.І., Рак Т.Є. Програмування мовою C++ : навчальний посібник. - Львів : Вид-во Львівського ДУ БЖД, 2011. - 292 с.
 8. Ю. А. Белов, Т. О. Карнаух, Ю. В. Коваль, А. Б. Ставровський. Вступ до програмування мовою C++. Організація обчислень : навч. Посіб - К.: Видавничополіграфічний центр "Київський університет 2012. - 175 с.
 9. [C++ common knowledge : essential intermediate programming/ C++ \(Computer program language\)](#), Dewhurst, Stephen C. Addison-Wesley, Upper Saddle River, N. J.: 2005.
 10. [C++ programming cookbook Herb Schildt's C++ programming cookbook / C++ \(Computer program language\)](#), Schildt, Herbert. McGraw-Hill, New York: c2008.
 11. [Problem solving with C++: The object of programming/ C++ \(Computer program language\)](#). Savitch, Walter. Pearson Addison Wesley, Boston: 2005. Fifth Edition (International ed.)
 12. [C++ programming : From Problem Analysis to Program Design / C plus plus programming.](#) : Malik, D S. Course Technology, Boston, MA : c2009. Fourth Edition.
 13. [Problem solving with C++ /](#) Savitch, Walter J, 1943- Pearson/Addison-Wesley, Boston : c2006.Sixth Edition.

Інформаційні ресурси

1. Google C++ Style Guide [Електронний ресурс]. – Режим доступу: <https://google.github.io/styleguide/cppguide.html>
2. The Stanford University C++ Style Guide [Електронний ресурс]. – Режим доступу: <https://hownot2code.com/2017/01/18/the-stanford-university-c-style-guide/>
3. 7 Best C++ Books For Beginners to Advanced Programmers Guide [Електронний ресурс]. – Режим доступу: <https://levelup.gitconnected.com/7-best-c-books-for-beginners-to-advanced-programmers-c7356471cad8>

Додатково

Методичне забезпечення

- **Сопронюк Т.М.** Об'єктно-орієнтоване програмування на C++ : навч. посібник / Т. М. Сопронюк. – Чернівці : Чернівецький національний ун-т, 2013. – 175 с. (з грифом МОНУ, лист №1/11-17600 від 18.11.2013)
- **Sopronyuk T.** Object-oriented programming in C++: Textbook / Translated by Nonna Shulga: CreateSpace, 2014. – 130 p. ISBN-13: 978-1502520906 (переклад навчального посібника з грифом МОНУ: Сопронюк Т.М. Об'єктно-орієнтоване програмування на C++ : навч. посібник / Т. М. Сопронюк. – Чернівці : Чернівецький національний ун-т, 2014. – 176 с.) <https://dspace.chnu.edu.ua/items/fe667704-8e73-4f3f-a69e-6b82dde9ac9c>
- [150 C++ Programming Assignments. Variants of tasks & Examples of the code](#): Textbook / Authored by Tatyana Sopronyuk, Translated by Nonna Shulga: CreateSpace, 2015. – 73 p. ISBN-13: 978-1515254065 (CreateSpace-Assigned) ISBN-10: 1515254062. <https://dspace.chnu.edu.ua/items/c8ff191f-5219-4311-bafe-d70298c21862>

- **Sopronyuk T.** C++ Programming: Theory and Assignments // Authored by Tatyana Sopronyuk, Translated by Nonna Shulga:Independently published, 2020. – 178 p. ISBN-13: 979-8651045624 , https://www.amazon.sg/Programming-Theory-Assignments-Tetyana-Sopronyuk/dp/B089M41MRX/ref=pd_rhf_se_p_img_2?encoding=UTF8&psc=1&refRID=RA189GRWAN2N4Y38HTB5
- Тестові завдання (Система Moodle)
- Презентації лекцій (Система Moodle)
- Відео-лекції на Google диску

Тематика задач контрольної роботи

1. *Робота з масивами структур (елементи структури - лише члени-дані, функції для роботи зі структурними змінними - зовнішні).*
2. *Створення класу та його використання. Клас включає члени-дані, конструктори, деструктор, функції-члени і функції-друзі класу, операції-члени і операції-друзі класу.*

Теоретичні питання до підсумкового тестування

1. Нові можливості C++, які не пов'язані з ООП: доступ до глобальних змінних, які закриті локальними, модифікатор const, функціональний і операторний записи перетворення типів.
2. Нові можливості C++, які не пов'язані з ООП: прототипи функцій, аргументи по замовчанню, посилання.
3. Нові можливості C++, які не пов'язані з ООП: функції, що підставляються, оператори динамічного розподілу пам'яті.
4. Нові можливості C++, які не пов'язані з ООП:перезавантаження функцій, шаблони функцій.
5. Нові можливості C++, які не пов'язані з ООП: перезавантаження операторів для структур (без <<, >>).
6. Нові можливості C++, які не пов'язані з ООП: перезавантаження операцій потокового вводу і виводу для структур (">>", "<<").
7. Інкапсуляція. Класи. Специфікатори public, private. Доступ до членів класу.
8. Статичні члени-дані, статичні члени-функції.
9. Дружні і складові (члени-функції) функції. Дружні класи.
10. Перезавантаження операторів для класів. Бінарні і унарні операції. Дружні оператори.
11. Перезавантаження операторів для класів: присвоєння і ініціалізація.
12. Перезавантаження операторів для класів: індексування, операція виклику функції.
13. Ініціалізація і знищення, конструктори і деструктори.
14. Стандартні об'єкти потокового вводу-виводу C++. Перезавантаження операцій "<<", ">>" для класу користувача.
15. Автоматичне і динамічне виділення пам'яті під об'єкти класу. Функції доступу до захованих членів класу.
16. Шаблони класів.
17. Утворення нових класів з існуючих. Агрегація. Порядок виклику конструкторів, і деструкторів при агрегації.

18. Успадкування. Похідні класи. Множинне успадкування.
19. Ієрархія класів. Специфікатори доступу. Співвідношення атрибутів доступу в базовому і похідному класах.
20. Успадкування: віртуальні базові класи.
21. Успадкування: перетворення вказівників на об'єкти класів.
22. Статичний та динамічний поліморфізм. Поліморфні класи.
23. Віртуальні функції-члени. Їх відмінність від звичайних функцій-членів.
24. Віртуальні деструктори. Поліморфні функції.
25. Абстрактні класи.
26. Огляд можливостей бібліотек STL, BOOST, MTL.
27. Робота з рядками засобами стандартної бібліотеки та без них.
28. Узагальнене програмування і стандартна бібліотека шаблонів STL. Приклади використання бібліотеки STL
29. Поняття компонентного класу. Бібліотека VCL
30. Побудова віконних інтерфейсів. Використання стандартних елементів керування.

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-yuriiia-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-vyjavlennia-ta-zapobihannia-akademichnomu-plahiatu-u-chernivetskomu-natsionalnomu-universyteti-imeni-yuriiia-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-yuriiia-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.