

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



РОБОЧА ПРОГРАМА
навчальної дисципліни

Об’єктно-орієнтоване програмування

(назва навчальної дисципліни)

Обов’язкова навчальна дисципліна

(вказати: обов’язкова / вибіркова)

Освітньо-професійна програма Системний аналіз

(назва програми)

Спеціальність 124 – Системний аналіз

(вказати: код, назва)

Галузь знань 12 – 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 2025

Мета навчальної дисципліни: поглиблене оволодіння сучасними технологіями програмування на С++ 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 С++ (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.

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 до абстрактного мислення, аналізу та синтезу

ЗК02. Здатність застосовувати знання у практичних ситуаціях

ЗК03. Здатність планувати і управляти часом

ЗК04. Знання та розуміння предметної області та розуміння професійної діяльності

ЗК05. Здатність спілкуватися державною мовою усно і письмово

ЗК10. Здатність працювати автономно

ЗК11. Здатність генерувати нові ідеї (креативність)

ЗК14. Здатність оцінювати та забезпечувати якість виконуваних робіт

фахові компетентності:

ФК1. Здатність використовувати системний аналіз як сучасну міждисциплінарну методологію, що базується на прикладних математичних методах та сучасних інформаційних технологіях і орієнтована на вирішення задач аналізу і синтезу технічних, економічних, соціальних, екологічних та інших складних систем.

ФК2. Здатність формалізувати проблеми, описані природною мовою, у тому числі за допомогою математичних методів, застосовувати загальні підходи до математичного моделювання конкретних процесів.

ФК6. Здатність до комп'ютерної реалізації математичних моделей реальних систем і процесів; проектувати, застосовувати і супроводжувати програмні засоби моделювання, прийняття рішень, оптимізації, обробки інформації, інтелектуального аналізу даних.

ФК7. Здатність використовувати сучасні інформаційні технології для комп'ютерної реалізації математичних моделей та прогнозування поведінки конкретних систем а саме: об'єктно-орієнтований підхід при проектуванні складних систем різної природи, прикладні математичні пакети, застосування баз даних і знань.

ФК8. Здатність організувати роботу з аналізу та проектування складних систем, створення відповідних інформаційних технологій та програмного забезпечення.

ФК9. Здатність представляти математичні аргументи і висновки з них з ясністю і точністю і в таких формах, які підходять для аудиторії як усно так і в письмовій формі.

ФК10. Здатність розробляти експериментальні та спостережувальні дослідження і аналізувати дані, отримані в них.

та отримуються наступні **програмні результати навчання:**

ПР8. Володіти сучасними методами розробки програм і програмних комплексів та прийняття оптимальних рішень щодо складу програмного забезпечення, алгоритмів процедур і операцій.

ПР9. Вміти створювати ефективні алгоритми для обчислювальних задач системного аналізу та систем підтримки прийняття рішень.

ПР12. Застосовувати методи і засоби роботи з даними і знаннями, методи математичного, логіко-семантичного, об'єктного та імітаційного моделювання, технології системного і статистичного аналізу.

ПР13. Проектувати, реалізовувати, тестувати, впроваджувати, супроводжувати, експлуатувати програмні засоби роботи з даними і знаннями в комп'ютерних системах і мережах.

ПР15. Розуміти українську та іноземну мови на рівні, достатньому для обробки фахових інформаційно-літературних джерел, професійного усного і письмового спілкування, написання текстів за фаховою тематикою.

Опис навчальної дисципліни Загальна інформація

Форма навчання	Рік підготовки	Семестр	Кількість			Кількість годин						Вид підсумкового контролю
			кредитів	годин	змістових	модулів лекцій	практичні	семінарські	лабораторні	самостійна робота	індивідуальні завдання	
Денна	2	1	5	150	3	30	-	-	30	74	16	Залік

Структура змісту навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	денна форма					
	усього	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	3		4	2	10

<p>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.</p>	31	6		6	4	15
Разом за змістовним модулем 1	50	9		10	6	25
Змістовий модуль 2. Інкапсуляція. Поняття класу						
<p>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 захищених 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.</p>	22	6		4	2	10
<p>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.</p>	28	5		6	2	15
Разом за змістовним модулем 2	50	11		10	4	25
Змістовий модуль 3. Успадкування і поліморфізм						
<p>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.</p>	14	3		4	2	8
<p>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.</p>	18	3		3	2	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 бібліотеки шаблонів STL. Приклади використання бібліотеки STL	18	4		3	2	8
Разом за змістовним модулем 3	50	10		10	6	24
Усього годин	150	30		30	16	74

Теми лабораторних занять

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

Індивідуальні завдання

Наведені в системі електронного навчання.

Самостійна робота

№ п/п	Назва теми	Кількість годин /форми контролю
1	Виняткові ситуації 1. Опрацювати теоретичні основи механізму обробки виняткових ситуацій у мові C++: ключові слова try, catch, throw, стандартні класи винятків. 2. Розробити програму мовою C++, у якій реалізувати обробку помилок введення даних за допомогою винятків. 3. Проаналізувати приклади використання власних класів винятків та підготувати короткий письмовий опис їх застосування у лабораторних роботах.	16 год Застосування в лабораторних роботах

2	Робота з потоками 1. Вивчити принципи роботи з потоками введення та виведення в C++ (cin, cout, cerr, файлові потоки). 2. Реалізувати програму, що здійснює зчитування даних з файлу та запис результатів обробки у файл із використанням потоків. 3. Пояснити особливості перевантаження операторів << та >> для класів користувача та їх застосування у власних програмах.	16 год Застосування в лабораторних роботах,
3	Графічні можливості 1. Ознайомитися з основними підходами до реалізації графічних об'єктів у C++ та їх використання в об'єктно-орієнтованому програмуванні. 2. Розробити програму, що описує ієрархію графічних фігур з використанням базового та похідних класів. 3. Проаналізувати застосування поліморфізму при роботі з графічними об'єктами та підготувати приклади для використання в лабораторних роботах.	22 год Застосування в лабораторних роботах
4	Створення віконних інтерфейсів (GUI) 1. Вивчити основні принципи побудови графічного інтерфейсу користувача та стандартні елементи керування (кнопки, поля введення, меню). 2. Реалізувати віконний інтерфейс для однієї з лабораторних робіт із використанням відповідної бібліотеки. 3. Описати структуру програми з графічним інтерфейсом та пояснити взаємодію між логікою програми і віконними елементами.	20 год Застосування в лабораторних роботах

Методи навчання:

- вербальні методи (лекція, бесіда, диспут, пояснення, розповідь тощо);
- практичні методи (практичні роботи);
- наочні методи (демонстрація, ілюстрація);
- робота з інформаційними ресурсами: з навчально-методичною, науковою, нормативною літературою та інтернет-ресурсами;
- самостійна робота над індивідуальним завданням або за програмою навчальної дисципліни;
- онлайн навчання з використанням відповідних онлайн-платформ.

Система контролю та оцінювання

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

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.
4. Нові можливості C++, які не пов'язані з ООП: функції, що підставляються, оператори динамічного розподілу пам'яті.
5. 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.

Критерії оцінювання результатів навчання з навчальної дисципліни

Захист лабораторної роботи та критерії оцінювання

- Задача 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 забов'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 заліку

Залік проводиться засобами 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 15 балів.
- У сумі з лабораторними роботами (60 балів) це загалом складатиме 100 балів.

Шкала оцінювання: національна та ЄКТС

100-бальна шкала	Оцінка за національною шкалою		Оцінка за шкалою ЄКТС	
			Оцінка	Пояснення за розширеною шкалою
90-100	Зараховано	Відмінно	A	відмінно
80-89		Добре	B	дуже добре
70-79			C	добре
60-69		Задовільно	D	задовільно
50-59			E	достатньо

35-49			FX	(незадовільно) з можливістю повторного складання
1-34	Незараховано	Незадовільно	F	(незадовільно) з обов'язковим самостійним повторним опрацюванням освітнього компонента до перескладання

Розподіл балів, які отримують студенти

Поточне тестування та самостійна робота							Залік	Сума
Змістовий модуль №1		Змістовий модуль № 2		Змістовий модуль № 3				
T1	T2	T3	T4	T5	T6	T7	40	100
6	5	8	10	13	11	7		

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

Зарахування результатів неформальної/інформальної освіти

Здобувачі вищої освіти має право на участь у неформальній/інформальній освіті.

У межах поточного контролю можуть визнаватися результати неформальної/інформальної освіти за умови наявності сертифікату або освітньої декларації про результати неформальної/інформальної освіти з питань, що відповідає тематиці курсу («Порядок визнання у Чернівецькому національному університеті імені Юрія Федьковича результатів навчання, здобутих шляхом неформальної та/або інформальної освіти», <https://www.chnu.edu.ua/media/4g5fzssb/poriadok-vyznannia-rezultativ-navchannia-zdobutykh-shliakhom-neformalnoi-ta-abo-informalnoi-osvity.pdf>).

Студентам можуть бути зараховані додаткові бали, отримані через неформальну освіту, до загальної суми балів, набраної з освітньої компоненти, за умови, що результати з проблеми, за якою відбувалося навчання, відповідають тематиці курсу.

Рекомендована література

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. 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. Глинський Я.М., Анохін В.Є., Рязька В.А. C++ і C++ Builder. -Львів: Деол, 2003.- 192 с.
5. Ткачук В.М. Програмування на C++: Лабораторний практикум. Видавництво Прикарпатського національного університету імені Василя Стефаника, 2011.- 160с
6. Грицюк Ю.І., Рак Т.Є. Програмування мовою C++ : навчальний посібник. - Львів : Вид во Львівського ДУ БЖД, 2011. - 292 с.
7. 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 уні верситет 2012. - 175 с.

8. Ярошко С.А. Методи розробки алгоритмів. Програмування мовою С++: Навчальний посібник / С.А. Ярошко, О.С. Ярошко – Львів: ЛНУ імені Івана Франка, 2022. – 248 с. [електронна версія: <https://lnuittutor.github.io/>]
9. Online compiler and debugger for c/c++ [Електронний ресурс] Режим доступу: <https://www.onlinegdb.com>
10. C++ reference [Електронний ресурс]. Режим доступу: <https://en.cppreference.com/w> 11. Learn C++ [Електронний ресурс]. Режим доступу: <https://www.learncpp.com/> 12. C++ Stories Stay up-to-date with Modern C++ [Електронний ресурс]. Режим доступу: <https://www.cppstories.com/>

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

1. **Сопронюк Т.М.** Об'єктно-орієнтоване програмування на С++ : 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.** 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 С++ : навч. посібник / Т. М. Сопронюк. – Чернівці : Чернівецький національний ун-т, 2014. – 176 с.)
3. **Sopronyuk T.** 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. – 177 p. ISBN-13: 979-8651045624
5. Тестові завдання (Система Moodle)
6. Презентації лекцій (Система Moodle)
7. Відео-лекції на Google диску

Політика академічної доброчесності

Дотримання політики щодо академічної доброчесності учасниками освітнього процесу при вивченні навчальної дисципліни регламентовано такими документами:

1. «Етичний кодекс Чернівецького національного університету імені Юрія Федьковича» <https://www.chnu.edu.ua/universitytet/normatyvni-dokumenty/etychnyi-kodeks-chernivetskoho-natsionalnoho-universytetu-imeni-yurii-a-fedkovycha/>
2. «Положенням про виявлення та запобігання академічного плагіату у Чернівецькому національному університету імені Юрія Федьковича» <https://www.chnu.edu.ua/universitytet/normatyvni-dokumenty/polozhennia-pro-vyavleniia-ta-zapobihanniia-akademichnomu-plahiatu/>