

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВЕТЕРИНАРНОЇ МЕДИЦИНИ ТА
БІОТЕХНОЛОГІЙ ІМЕНІ С.З. ГИЖИЦЬКОГО

Факультет механіки, енергетики та інформаційних технологій
Кафедра інформаційних технологій

ПОГОДЖЕНО

Гарант ОПП «Комп'ютерні науки»
Вадим ПТАШНИК

(ім'я та прізвище, підпис)

«27» серпня 2025 року

ЗАТВЕРДЖЕНО

Декан факультету механіки, енергетики та
інформаційних технологій
Степан КОВАЛИШИН

(ім'я та прізвище, підпис)

«28» серпня 2025 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«ПОГЛИБЛЕНЕ ВИВЧЕННЯ АЛГОРИТМІЗАЦІЇ»

(код і назва навчальної дисципліни)

рівень вищої освіти перший (бакалаврський)
(назва освітнього рівня)

галузь знань 12 «Інформаційні технології»
(назва галузі знань)

спеціальність 122 «Комп'ютерні науки»
(назва спеціальності)

освітня програма «Комп'ютерні науки»
(назва)

вид дисципліни вибіркова
(обов'язкова / за вибором)

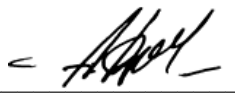
2025–2026 навчальний рік

Робоча програма «Поглиблене вивчення алгоритмізації»
(назва навчальної дисципліни)

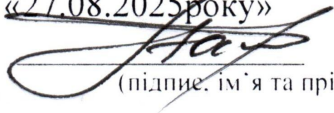
Укладач: Заплатинський Н.Б., старший викладач кафедри інформаційних технологій
(вказати укладачів, їхні посади, наукові ступені та вчені звання)

Робочу програму схвалено на засіданні кафедри інформаційних технологій.

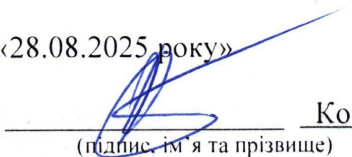
Протокол № 1 від 26.08.2025 року.

Завідувач кафедри  Тригуба А.М.
(підпис, ім'я та прізвище)

Погоджено навчально-методичною комісією спеціальності
«Комп'ютерні науки»
(назва спеціальності)

Протокол № 8/1 від «27.08.2025 року»
Голова НМКС  Пташник В.В.
(підпис, ім'я та прізвище)

Схвалено рішенням навчально-методичної ради факультету ФМЕІТ
(назва факультету)

Протокол № 1 від «28.08.2025 року»
Голова НМРФ  Ковалишин С.Й.
(підпис, ім'я та прізвище)

Ухвалено вченою радою ФМЕІТ протокол №1 від 28.08.2025 р.

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Всього годин	
	денна форма здобуття освіти	заочна форма здобуття освіти
Семестр	8	8
Кількість кредитів/годин	4/120	4/120
Усього годин аудиторної роботи	36	14
в т.ч.:		
• лекційні заняття, год.	12	6
• практичні заняття, год.	-	-
• лабораторні заняття, год.	24	8
• семінарські заняття, год.	-	-
Усього годин самостійної роботи	84	106
Форма контролю	іспит	іспит

Примітка.

Частка аудиторного навчального часу студента у відсотковому вимірі:

для денної форми здобуття освіти – 30%

для заочної форми здобуття освіти – 11,6 %

1. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Метою вивчення освітньої компоненти "Поглибленне вивчення алогоритмізації" є навчити студента:

- оцінювати різні типи алгоритмів стосовно їх часової та ємнісної складності;
- використовувати різні типи композицій алгоритмів для побудови ефективних алгоритмів розв'язування практично важливих задач;
- використовувати класичні алгоритмічні системи (нормальні алгоритми Маркова, рекурсивні функції, машини Тьюрінга) для аналізу розв'язності різних типів задач;
- на основі відповідних підходів (метод „поділяй і володарюй”, ”жадібні алгоритми”, динамічне програмування) розробляти та реалізовувати алгоритми для конкретних задач.

Завдання навчальної дисципліни передбачають: дати змогу майбутнім фахівцям у сфері інформаційних технологій ефективно проектувати, розробляти та застосовувати сучасні методи програмування, оптимізації програмного забезпечення.

Міждисциплінарні зв'язки: освітня компонента «Поглиблене вивчення алогоритмізації» є складовою частиною циклу професійної підготовки для здобувачів освітньо-професійної програми «Комп'ютерні науки» першого (бакалаврського) рівня вищої освіти. Вивчення дисципліни передбачає наявність систематичних та ґрунтовних знань із суміжних курсів – «Алгоритми та структури даних», «Програмування», «Інформаційні технології».

Мета навчальної дисципліни. Метою вивчення навчальної дисципліни є формування у студентів теоретичних знань і практичних навичок в області алгоритмів. Ціллю вивчення навчальної дисципліни є сформулювати у студентів теоретичних знань та практичних навичок, необхідних для створення застосунків.

Прековізити: освітня компонента «Поглиблене вивчення алогоритмізації» є вибірковою компонентою, яка забезпечує професійну підготовку здобувачів освітньо-професійної програми «Комп'ютерні науки» першого (бакалаврського) рівня вищої освіти. Вивчення дисципліни передбачає наявність систематичних та ґрунтовних знань із суміжних курсів першого (бакалаврського) рівня вищої освіти – «Програмування», «Алгоритми та структури даних», «Комп'ютерні мережі», «Бази даних».

Пострековізити: вивчення дисципліни «Поглиблене вивчення алогоритмізації» створює підґрунтя для опанування наступних компонент освітньої програми. Отримані знання та

компетентності особливо важливі під час проходження практики, виконання курсових та кваліфікаційної роботи, що сприяє формуванню професійних умінь у сфері проектування та програмного забезпечення.

Результати навчання: у результаті вивчення дисципліни «Поглиблене вивчення алогоритмізації» студенти оволодівають здатністю формулювати алгоритмічні моделі задач, будувати та аналізувати алгоритми з урахуванням їхньої ефективності, програмно реалізовувати обрані рішення та проводити їхню перевірку. Вони вчаться зіставляти різні алгоритмічні підходи, обґрунтовувати вибір найбільш доцільного методу та застосовувати набуті знання для розв'язання практичних і дослідницьких завдань у сфері комп'ютерних наук..

2. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви тем	Кількість годин											
	денна форма						заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб.	інд.	с.р.		л	п	лаб.	інд.	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
	Семестр 8						Семестр 8					
Тема 1.	8	1		2		5	8					8
Тема 2.	11	1		4		6	11	2		1		8
Тема 3.	9	1		2		6	9					9
Тема 4.	11	2		2		7	11	1		1		9
Тема 5.	11	1		2		8	11			1		10
Тема 6.	11	2		2		7	11	1		1		9
Тема 7.	11	2		2		7	11	1		2		8
Тема 8.	9	1		4		4	9			2		7
Тема 9.	9	1		4		4	9	1				8
Іспит	30	-	-	-	-	30	30	-	-	-	-	30
Усього годин	120	12		24		84	120	6		8		106

3. ЛЕКЦІЙНІ ЗАНЯТТЯ

№ з/п	Назви тем та їх короткий зміст	Кількість годин	
		ДФЗО	ЗФЗО
1	Тема 1. Базові поняття алгоритмів та їх складності. Методи оцінювання та побудова алгоритмів із заданими оцінками. Розробка й аналіз алгоритмів.	1	
2	Тема 2. Алфавітні оператори та властивості алгоритмів. Дослідження часової та ємнісної складності алгоритмів сортування.	1	2
3	Тема 3. Способи композиції алгоритмів і граф-схеми. Побудова алгоритмів із заданими оцінками.	1	
4	Тема 4. Класичні алгоритмічні системи. Система нормальних алгоритмів Маркова, універсальний нормальний алгоритм. Приклади композиції алгоритмів.	2	1

5	Тема 5. Рекурсивні функції: найпростіші функції та способи їх композиції.	1	
6	Тема 6. Оператори суперпозиції та примітивної рекурсії. Примітивно-рекурсивні функції та найпростіші приклади рекурсивних алгоритмів.	2	1
7	Тема 7. Оператор мінімізації та частково-рекурсивні функції. Суперпозиція і примітивна рекурсія як інструменти побудови алгоритмів.	2	1
8	Тема 8. Машини Тюрінга: варіанти побудови, детерміновані машини, теза Тюрінга. Застосування оператора мінімізації в алгоритмах.	1	
9	Тема 9. Зв'язок машин Тюрінга з іншими алгоритмічними системами. Побудова машин Тюрінга для типових алгоритмів.	1	1
Усього годин		12	6

4. ЛАБОРАТОРНІ ЗАНЯТТЯ

№ з/п	Назви тем та їх короткий зміст	Кількість годин	
		ДФЗО	ЗФЗО
1	Аналіз базових алгоритмів. Реалізація та дослідження часової і просторової складності алгоритмів пошуку та сортування.	2	
2	Композиція алгоритмів. Побудова граф-схем для комбінованих алгоритмів і програмна реалізація складених обчислювальних процесів.	4	1
3	Нормальні алгоритми Маркова. Розробка алгоритмів для роботи з формальними мовами та перевірка їх коректності.	2	
4	Рекурсивні функції. Реалізація рекурсивних алгоритмів для класичних задач	2	1
5	Машина Тюрінга. Моделювання роботи детермінованої машини Тюрінга для розв'язання простої задачі (наприклад, додавання двійкових чисел).	2	1
6	Недетерміновані алгоритми. Побудова прикладів недетермінованих машин Тюрінга для окремих задач та експериментальне порівняння з детермінованими моделями.	2	1
7	NP-повні задачі. Практичний аналіз задачі кліки та задачі комівояжера на малих графах з оцінюванням складності та обговоренням методів оптимізації.	2	2
8	Машини Тюрінга: варіанти побудови, детерміновані машини, теза Тюрінга. Застосування оператора мінімізації в алгоритмах.	4	2
9	Зв'язок машин Тюрінга з іншими алгоритмічними системами. Побудова машин Тюрінга для типових алгоритмів.	4	
Усього годин		24	8

5. САМОСТІЙНА РОБОТА

№ з/п	Назви тем та їх короткий зміст	Кількість годин	
		ДФЗО	ЗФЗО
1	Оцінка складності алгоритмів. Теоретичний аналіз і підготовка таблиці порівняння часової та просторової складності алгоритмів сортування.	5	8

2	Властивості алгоритмів. Опрацювання властивостей (дискретність, детермінованість, масовість) та написання короткого есе з прикладами.	6	8
3	Алгоритмічні системи. Огляд ідей алгоритмів Маркова й машин Поста як прикладів класичних систем обчислень.	6	9
4	Рекурсивні функції. Теоретичне доведення властивостей примітивно-рекурсивних і частково-рекурсивних функцій.	7	9
5	Теорія обчислюваності. Опис та аналіз прикладів застосування машини Тюрінга для задач із різних класів (поліноміальні, експоненційні).	8	10
6	Важкорозв'язні задачі. Підготовка реферату про сучасні методи розв'язання NP-повних задач та підхід «наближених алгоритмів».	7	9
7	Ефективні алгоритми. Огляд сучасних методів оптимізації алгоритмів та порівняння їх з класичними методами, представленими на курсі.	7	8
8	Машини Тюрінга: варіанти побудови, детерміновані машини, теза Тюрінга. Застосування оператора мінімізації в алгоритмах.	4	7
9	Зв'язок машин Тюрінга з іншими алгоритмічними системами. Побудова машин Тюрінга для типових алгоритмів.	4	8
Усього годин		84	106

6. ІНДИВІДУАЛЬНІ ЗАВДАННЯ

—

7. МЕТОДИ НАВЧАННЯ

Навчання з дисципліни «Поглиблене вивчення алгоритмізації» здійснюється із застосуванням сучасних практикоорієнтованих методів, що поєднують словесні (лекція, пояснення, дискусія), наочні (демонстрація, робота з мультимедійними матеріалами) та практичні форми (комплексні задачі, лабораторні роботи).

8. МЕТОДИ КОНТРОЛЮ

Оцінювання результатів навчання студентів здійснюється проведенням поточного та підсумкового контролю.

Поточний контроль здійснюється під час практичних занять і має на меті перевірку рівня підготовленості студента до виконання відповідних завдань. Форми проведення поточного контролю – усне та письмове опитування, тестовий контроль.

Підсумковий контроль проводиться з метою оцінювання результатів навчання на завершальному етапі вивчення дисципліни. Підсумковий контроль здійснюється у формі екзамену.

9. КРИТЕРІЇ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ

Успішність студента оцінюється шляхом проведення поточного та підсумкового контролю.

Максимальна кількість балів з дисципліни «Поглиблене вивчення алгоритмізації», яку може отримати студент протягом семестру за всі види роботи, становить 100, при цьому 50 балів за результатами поточного оцінювання, та 50 – за результатами екзаменаційного контролю.

Результати **поточного контролю** оцінюються за чотирибальною («2», «3», «4», «5») шкалою. В кінці семестру обчислюється середнє арифметичне значення (САЗ) усіх отриманих студентом оцінок з наступним переведенням його у 50-ти бальну шкалу за формулою:

$$ПК = 10 \cdot САЗ$$

Критерії поточного оцінювання знань студентів

Оцінка	Критерії оцінювання
5 («відмінно»)	У повному обсязі володіє навчальним матеріалом, вільно, самостійно та аргументовано його викладає, глибоко і всебічно розкриває зміст, використовуючи обов'язкову та додаткову літературу. Правильно вирішив 90% тестових завдань.
4 («добре»)	Достатньо повно володіє навчальним матеріалом, обґрунтовано його викладає, в основному розкриває зміст завдань, використовуючи обов'язкову літературу. При викладанні окремих питань не вистачає достатньої глибини та аргументації, допускаються несуттєві неточності й незначні помилки. Правильно вирішив більшість тестових завдань.
3 («задовільно»)	У цілому володіє навчальним матеріалом, викладає його основний зміст, але без глибокого всебічного аналізу, обґрунтування та аргументації, допускаючи окремі суттєві неточності та помилки. Правильно вирішив близько половини тестових завдань.
2 («незадовільно»)	Не в повному обсязі володіє навчальним матеріалом. Викладає матеріал фрагментарно та поверхово, без аргументації й обґрунтування, недостатньо розкриває зміст теоретичних і практичних завдань, допускає суттєві неточності. Правильно вирішив меншість тестових завдань.

Переведення підсумкових рейтингових оцінок з дисципліни, виражених у балах за 100-бальною шкалою, у оцінки за національною шкалою та шкалою ECTS

Таблиця 1 – **Шкала оцінювання: національна та ECTS**

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, диференційованого заліку, курсового проекту (роботи), практики	для заліку
90–100	A	відмінно	зараховано
82–89	B	добре	
74–81	C		
64–73	D	задовільно	
60–63	E		
35–59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання

0–34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни
------	----------	--	---

10. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Підручники і навчальні посібники; інструктивно-методичні матеріали до практичних занять; індивідуальні навчально-дослідні завдання; контрольні роботи; текстові та електронні варіанти тестів для поточного контролю, методичні матеріали для організації самостійної роботи студентів, виконання індивідуальних завдань.

11. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Базові

1. Клакович Л.М., Левицька С.М., Костів О.В. Теорія алгоритмів. Львів, Вид-во Львів ун-ту, 2014.- 116 с.
2. Троцько В.В. Теорія алгоритмів: Навчально - методичний посібник. – Київ: Університет економіки та права «КРОК», 2023 – 123 с.
3. Томас Кормен, Чарльз Лейзерсон, Рональд Рівест, Кліффорд Стайн. Вступ до алгоритмів. К: – К.І.С., 2019. – 1288 с.
4. Erich Gamma. Design Patterns: Elements of Reusable Object-Oriented Software / Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Grady Booch. – Addison-Wesley Professional, 1994. – 416 p. ISBN-10: 0201633612, ISBN-13: 978-0201633610.
5. Edward Sciore . Java Program Design: Principles, Polymorphism, and Patterns. Apress, 2019. - 465 Pages
6. Jason R. Briggs. Python for kids. A Playful Introduction to Programming - No Starch Press, Inc., San Francisco, CA 2013 – 318p

Допоміжна

7. M. Lutz: Learning Python: Powerful Object-Oriented Programming, 5th ed. // O'Reilly Media, Inc., 2013.
8. D. Beazley, B.K. Jones: Python Cookbook: Recipes for Mastering Python 3, 3rd ed. // O'Reilly Media, Inc., 2013.
9. A. Martelli, A. Ravenscroft, S. Holden: Python in a Nutshell: The Definitive Reference, 3rd ed. // O'Reilly Media, Inc., 2017.
10. B. Lubanovic: Introducing Python: Modern Computing in Simple Packages. // O'Reilly Media, Inc., 2015.
11. J. Hunt: A Beginners Guide to Python 3 Programming. // Springer, 2019.
12. J. Hunt: Advanced Guide to Python 3 Programming. // Springer, 2019

12. Інформаційні ресурси

1. Бібліотечно-інформаційні ресурси – книжковий фонд, періодика та фонди на електронних носіях бібліотеки ЛНУВМБ, державних органів науково-технічної інформації, наукових, науково-технічних бібліотек та інших наукових бібліотек України.

2. Електронний ресурс стосовно ІТ компаній Львівського регіону:
http://www.invest-lvivregion.com/it-компанії_ua_285cms.htm

3. ІТ компанії Львова:
http://it-catalogue.net/ru/component/companies_cat/companies/458/all/default/all/main.html

4. Портал об'єктно-орієнтованого програмування: <http://oop.in.ua/tag/FAQ/>

5. Книжки з програмування: як читати і що саме:
<https://dou.ua/lenta/articles/programming-books/>

6. Українська технічна література. Програмування:
<https://ukrtechlibrary.wordpress.com/tag/програмування/>

7. Задачі програмування із прикладами розв'язку: <http://purecodecpp.com/uk/archives/433>

8. Задачі програмування із прикладами розв'язку:
<http://library.nuft.edu.ua/ebook/datathree.php?ID=138>

9. Задачі програмування із прикладами розв'язку:
<http://abramov.org.ua/blog/category/opp/obchislennya-%D1%96z-zber%D1%96gannyam-posl%D1%96dovnosti/>