

КРИВОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
Кафедра моделювання та програмного забезпечення

ЗАТВЕРДЖУЮ

Перший проректор

_____ Владислав ЧУБАРОВ

“ _____ ” _____ 2025 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Дискретні структури

(шифр і назва навчальної дисципліни)

спеціальність

_____ 121 – «Інженерія програмного забезпечення»

(шифр і назва напряму підготовки)

факультет

_____ Інформаційних технологій

(назва інституту, факультету, відділення)

Форма навчання	Курс	Семестр	Всього годин за планом	Кількість кредитів	Всього аудиторних годин	Аудиторних годин, (у тому числі КЗ)		Самостійна робота (год.)	Контрольно-модульні роботи	Залік (сем.)	Екзамен (сем.)
						Лекції	Лабораторні				
Денна	2	4	210	7	90	54	36	120	2		*
Заочна	2	4	210	7	18	10	8	192	-		*

Робочу програму навчальної дисципліни «Дискретні структури» для здобувачів першого (бакалаврського) рівня вищої освіти за освітньою програмою «Інженерія програмного забезпечення» розроблено згідно з ОПП галузі знань 12 «Інформаційні технології» зі спеціальності 121 «Інженерія програмного забезпечення».

Розробники: канд. пед. наук, доцент Стрюк Андрій Миколайович

Робоча програма затверджена на засіданні кафедри моделювання та програмного забезпечення

Протокол від “ ” _____ 2024 року № ____

Завідувач кафедри МПЗ, доцент, к.п.н. _____ Андрій СТРЮК

Схвалено вченою радою факультету інформаційних технологій

Протокол від “ ” _____ 2024 року № ____

Голова вченої ради _____ Іван МУЗИКА

Схвалено групою забезпечення ОПП

Протокол від “ ” _____ 2024 року № ____

Гарант ОПП _____ Андрій СТРЮК

ЗМІСТ

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	4
2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	5
3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	7
4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ.....	8
5. ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ.....	9
6. ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ.....	9
7. САМОСТІЙНА РОБОТА.....	10
8. МЕТОДИ НАВЧАННЯ.....	11
9. МЕТОДИ КОНТРОЛЮ.....	12
10. РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ ЗДОБУВАЧІ.....	12
11. ПЕРЕЛІК ПИТАНЬ ДЛЯ ПІДСУМКОВОГО КОНТРОЛЮ ЗНАНЬ.	16
12. НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ З ДИСЦИПЛІНИ.....	17
13. ІНФОРМАЦІЙНІ РЕСУРСИ.....	17
14. ТЕРМІНОЛОГІЧНИЙ СЛОВНИК	18
Додаток до робочої програми. Робочий план.....	22

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Галузь знань, спеціальність, ступінь вищої освіти	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів – 7	Галузь знань <u>12 Інформаційні технології</u> (шифр і назва)	Нормативна	
Модулів – 1	Спеціальність <u>121 Інженерія програмного забезпечення</u> (код та найменування спеціальності)	Рік підготовки:	
Змістових модулів – 2		2	2
Загальна кількість годин - 210		Семестр	
		4-й	4-й
		Лекції	
Тижневих годин для денної форми навчання: аудиторних – 5 самостійної роботи студента – 6,67	Ступінь вищої освіти: <u>бакалавр</u>	54 год.	10 год.
		Практичні, семінарські	
		-	-
		Лабораторні	
		36 год.	8 год.
		Самостійна робота	
		120 год.	192 год.
Вид контролю: Екзамен			

Примітка.

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить:

- для денної форми навчання – 0,75
- для заочної форми навчання – 0,09

2. МЕТА ТА ЗАВДАННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

2.1. Метою дисципліни «Дискретні структури» є формування у здобувача розуміння основ дискретної математики, що є фундаментом для розуміння та розробки алгоритмів, структур даних, баз даних та комп'ютерних мереж, та здобуття практичних навичок використання базових дискретних структур в практичних ситуаціях як-то робота з множинами, комбінаторні задачі, кодування та задачі на графах.

2.2. Основними завданнями вивчення дисципліни «Дискретні структури» є:

- розгляд фундаментального поняття множини, відношень та операцій з множинами та елементами множин;
- вивчення поняття алгебраїчних структур та їх типів, особливостей булевої алгебри, логічних функцій та логічного числення;
- вивчення основ комбінаторики та її застосуванні в програмуванні, вивчення принципів кодування та двійкового представлення різних типів даних, оптимальне та завадостійке кодування;
- опанування основ теорії графів та практичне застосування фундаментальних алгоритмів на графах.

2.3. В результаті опанування дисципліни студенти освоюють наступні компетентності:

загальні

- ЗК02 Здатність застосовувати знання у практичних ситуаціях;

фахові

- СК07 Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних;
- СК08 Здатність застосовувати фундаментальні та міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення;
- СК14 Здатність до алгоритмічного та логічного мислення.

2.4. Програмні результати навчання освітньої програми, яким відповідає дисципліна:

- ПР05 Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення;
- ПР06 Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення;
- ПР13 Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

В результаті вивчення дисципліни студенти повинні *знати*:

- поняття множини, математичний опис відношень та операцій з множинами;
- властивості відношень між елементами множини;
- поняття алгебраїчної структури, типи алгебраїчних структур;
- булеві функції та логічне числення;
- базові комбінаторні задачі, принципи оцінки складності алгоритмів;
- основи теорії кодування, способи кодування різних типів даних, методи оптимального та завадостійкого кодування;
- основи теорії графів, поняття графу, орієнтованого графу, дерева, бінарного дерева;
- фундаментальні алгоритми на графах;

вміти:

- використовувати множини в програмах; вирішувати практичні задачі з обробкою множин;
- використовувати логічне числення в практичних ситуаціях;
- розв'язувати комбінаторні задачі, пов'язані з перебором різних комбінацій даних;
- використовувати на практиці різні способи кодування даних та алгоритми стиснення даних;
- працювати з різними представленнями графів та дерев;
- використовувати фундаментальні алгоритми на графах у практичних ситуаціях.

2.5. Міждисциплінарні зв'язки

При вивченні дисципліни використовуються знання здобувачів з дисциплін «Основи програмування», «Алгоритми та структури даних», «Вища математика».

Знання, одержані здобувачами при вивченні дисципліни, використовуються при вивченні дисциплін «Чисельні методи», «Безпека програм та даних», «Системи штучного інтелекту».

Вимоги до знань та умінь визначаються галузевими стандартами вищої освіти України.

3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Заліковий модуль 1 Дискретні структури

Змістовий модуль 1. Множини та алгебраїчні структури

Тема 1. Вступ до дисципліни дискретні структури – 2 год.

Дискретні структури як математичний фундамент програмування. Зв'язок дискретної математики з іншими дисциплінами підготовки майбутніх інженерів-програмістів. Ознайомлення зі структурою, цілями та задачами курсу.

Тема 2. Множини – 4 год.

Базові поняття. Множини та елементи множин. Потужність множин. Підмножини. Пуста множина та Універсум. Парадокс Рассела. Діаграми Ейлера та діаграми Ейлера-Вена як наочне зображення множин та відношень між ними. Використання множин в сучасних мовах програмування.

Тема 3. Відношення та операції над множинами – 4 год.

Відношення між множинами. Операції над множинами. Декартовий добуток двох множин.

Тема 4. Відношення між елементами множин – 4 год.

Бінарні відношення між елементами множин. Область визначення й область значень. Відповідності. Властивості відношень. Комп'ютерне подання відношень.

Тема 5. Алгебраїчні структури – 4 год.

Поняття алгебраїчної структури. Поняття операції. Властивості операції. Гомоморфізм. Морфізми. Типи алгебраїчних структур. Булева алгебра

Тема 6. Булеві функції – 4 год.

Функції алгебри логіки. Булеві функції однієї змінної. Суттєві та несуттєві змінні. Булеві функції двох змінних. Реалізація функцій формулами. Рівносильні формули.

Тема 7. Комбінаторика – 6 год.

Комбінаторні задачі. Перестановки. Розміщення. Поєднання. Підстановки.

Змістовий модуль 2. Теорія кодування. Графи та дерева

Тема 8. Кодування – 8 год.

Кодування як основний спосіб представлення даних. Аналогові та дискретні сигнали. Аналого-дискретне перетворення. Біт як одиниця інформації та даних. Різновиди даних. Кодування чисел. Кодування символів. Кодування зображень. Кодування звуку. Теорема Шеннона про оптимальне кодування. Алгоритм Шеннона-Фано. Алгоритм Хаффмана. Арифметичне кодування. Словникові алгоритми.

Тема 9. Графи – 8 год.

Витоки та основні визначення теорії графів. Елементи графа. Орієнтовані графи. Графи з петлями. Ізоморфізм графів. Підграфи. Доповнення графа. Спрямовані орграфи та мережі. Маршрути, ланцюги, цикли. Відстань між вершинами. Комп'ютерне представлення графів. Деревя. Визначення та основні властивості дерев. Орієнтовані дерева. Бінарні дерева. Комп'ютерне представлення дерев. Обходи бінарних дерев. Приклади використання дерев.

Тема 10. Фундаментальні алгоритми на графах – 10 год.

Обхід графа. Пошук в глибину. Пошук в ширину. Пошук найкоротшого шляху в графі. Алгоритм Дейкстри. Алгоритм Флойда-Воршалла. Побудова оптимального остового дерева. Алгоритм Крускала. Алгоритм Пріма.

4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви змістових модулів і тем	Кількість годин							
	Денна форма				Заочна форма			
	усього	у тому числі			усього	у тому числі		
		лекції	лабораторні	самостійна робота		лекції	лабораторні	самостійна робота
1	2	3	4	5	6	7	8	9
Заліковий модуль 1								
Змістовий модуль 1. Множини та алгебраїчні структури								
Тема 1. Вступ до дисципліни дискретні структури	4	2	-	2	4	-	-	4
Тема 2. Множини	14	4	2	8	14	1	-	13
Тема 3. Відношення та операції над множинами	18	4	4	10	18	1	1	16
Тема 4. Відношення між елементами множин	18	4	4	10	18	1	1	16
Тема 5. Алгебраїчні структури	10	4	-	6	10	1	-	9
Тема 6. Булеві функції	18	4	4	10	18	1	1	16
Тема 7. Комбінаторика	24	6	4	14	24	1	1	22
Разом за змістовий модуль 1	106	28	18	60	106	6	4	96
Змістовий модуль 2. Теорія кодування. Графи та дерева								
Тема 8. Кодування	32	8	6	18	32	1	1	30
Тема 9. Графи	34	8	6	20	34	1	1	32
Тема 10. Фундаментальні алгоритми на графах	38	10	6	22	38	2	2	34
Разом за змістовий модуль 2	104	26	18	60	104	4	4	96
Усього годин за семестр	210	54	36	120	210	10	8	192

5. ТЕМИ ПРАКТИЧНИХ ЗАНЯТЬ

Не передбачено.

6. ТЕМИ ЛАБОРАТОРНИХ ЗАНЯТЬ

№ з/п	Назва теми	Кількість годин	
		Денна ф. н.	Заочна ф. н.
1.	Множини	2	-
2.	Відношення та операції над множинами	4	1
3.	Відношення між елементами множин	4	1
4.	Булеві функції	4	1
5	Комбінаторика	4	1
6.	Кодування	6	1
7.	Графи	6	1
8.	Задачі на графах	6	2
РАЗОМ		36	8

7. САМОСТІЙНА РОБОТА

На самостійну роботу студентам денної форми навчання відведено 120 годин, заочної – 192 години.

Самостійна робота студентів при вивченні дисципліни «Дискретні структури» залучає такі складові:

- підготовка до лекцій через вивчення попереднього лекційного матеріалу, а також літературних джерел, на яких ґрунтується матеріал попередніх лекцій;
- вивчення окремих тем або питань, що передбачені для самостійного опрацювання;
- підготовка до лабораторних занять через вивчення теоретичного матеріалу, необхідного для відповіді на тестові запитання та готовності до вирішення завдань;
- виконання лабораторних робіт;
- підготовка до проведення контрольних заходів.

При виконанні самостійної роботи студент денної форми навчання повинен:

- скласти рішення завдань лабораторних робіт;
- оформити звіт з лабораторних робіт;

При виконанні самостійної роботи студент заочної форми навчання повинен:

- опрацювати необхідний теоретичний матеріал;
- скласти рішення завдань лабораторних робіт;
- оформити звіт з лабораторних робіт

Питання для самостійного опрацювання

№	Назва теми	Кількість годин	
		денна ф.н.	заочн а ф.н.
1.	Вивчить особливості реалізації множин в обраній вами мові програмування (Python, C++, Java, C# на вибір)	3	5
2.	Вивчить методи роботи з множинами (об'єднання, віднімання, перетин) в обраній вами мові програмування	3	5
3.	Дослідіть реалізацію логічних операцій в обраній вами мові програмування	3	5
4.	Кон'юнктивна нормальна форма (КНФ) та диз'юнктивна нормальна форма (ДНФ)	3	5
5.	Дослідіть популярні методи стиснення зображень та відео	3	5
6.	Розгляньте бібліотеки, що реалізують методи стиснення даних, доступні для обраної вами мови програмування	3	5
7.	Огляд бібліотек для роботи з графами для обраної вами мови програмування	3	5
8.	Оберіть та вивчіть засоби побудови графічних зображень для візуалізації графів в обраній вами мові програмування	3	5
9.	Розгляньте алгоритми створення лабіринтів	3	5
10.	Ознайомтесь з алгоритмами розв'язання лабіринтів	3	5
11.	Опрацювання матеріалу лекцій, робота з рекомендованою літературою	18	62
12.	Підготовка до лабораторних занять	18	36
13.	Виконання лабораторних робіт	36	36
14.	Підготовка звітів з лабораторних робіт	8	8
15.	Підготовка до контрольних заходів	10	-
	Разом	120	192

8. МЕТОДИ НАВЧАННЯ

Використовуються наступні методи навчання: лекції, лабораторні заняття, самостійна робота.

Навчальна лекція – це логічне, послідовне викладання змісту навчання, яке характеризується судженнями, висновками, підсумком. Вона охоплює основний теоретичний матеріал однієї або кількох тем навчальної дисципліни. Призначенням лекції є формування у здобувачів фундаментальних знань з дисципліни, а також визначає основний зміст і характер усіх інших навчальних занять та самостійної роботи здобувачів із цієї дисципліни.

Лабораторне заняття - форма організації навчання, яку проводять за завданням і під керівництвом НПП. Основні дидактичні цілі – експериментальне підтвердження вивчених теоретичних положень навчальної дисципліни та формування вмінь й навичок їх практичного застосування. Проведення лабораторного заняття ґрунтується на попередньо підготовлених наборах завдань різної складності для розв'язання на занятті. Лабораторне заняття проводиться у навчальних лабораторіях з використанням пристосованого до умов навчального процесу устаткування.

Самостійна робота здобувача є основним способом оволодіння навчальним матеріалом у час, вільний від обов'язкових аудиторних занять. Мета виконання самостійної роботи – поглиблення, узагальнення й закріплення теоретичних знань і практичних умінь здобувачів із дисципліни шляхом вироблення вмінь самостійної роботи з навчальною і фаховою літературою та інформацією в мережі Інтернет.

Самостійна робота здобувачів здійснюється у формі: підготовки до лекцій і лабораторних занять, виконанні самостійних проєктів. Самостійну роботу здобувач може виконувати у бібліотеці, комп'ютерних класах, а також у домашніх умовах.

Підготовка до лекцій передбачає самостійне опрацювання теоретичного матеріалу. При цьому необхідно звернути увагу на необхідність чіткого засвоєння основних термінів та визначень, розуміння їх змісту, обов'язкового аналізу використання теоретичних положень для розв'язання наданих прикладів.

Самоперевірку засвоєння навчального матеріалу здобувач здійснює за контрольними запитаннями, що надано після кожної теми у конспекті лекцій та іншій літературі, та після кожного лабораторного заняття у відповідних методичних вказівках. Якщо на деякі запитання здобувач не може надати відповіді, то необхідно повторити вивчення навчального матеріалу, або визначити правильну відповідь з викладачем на консультації.

Під час вивчення даної дисципліни використовуються:

- мультимедійні освітні технології: інтерактивні лекції (презентації) із використанням програми MS Power Point у поєднанні з анімацією та звуковим супроводом; перегляд відеороликів за окремими пунктами тем занять, використання електронних посібників;
- діалогові технології: організація групових обговорень, використання «мозкового штурму».

Лекції проводяться з використанням технічних засобів навчання й супроводжуються демонстрацією презентацій за допомогою проектора.

У разі виникнення необхідності забезпечення навчального процесу в дистанційному режимі супровід та контроль знань реалізовується за допомогою дистанційного курсу, розробленого в Google Classroom. Онлайн лекції, консультації та усні відповіді на питання, захист проєктів проводиться за допомогою Google Meet або Zoom.

9. МЕТОДИ КОНТРОЛЮ

Основними завданнями контролю знань здобувачів вищої освіти з дисципліни є оцінювання засвоєння теоретичних знань і практичних навичок, отриманих під час навчання.

Контрольні заходи мають виконувати наступні функції:

- стимулювати систематичну самостійну роботу над навчальним матеріалом;
- забезпечувати закріплення та реалізацію набутих теоретичних знань при підготовці до практичних занять;
- прищеплювати навички відповідального ставлення до своїх обов'язків, самостійного цілеспрямованого пошуку потрібної інформації, чіткої організації свого робочого дня.

Оцінювання знань здобувачів складається з поточного, модульного та підсумкового контролю.

Поточний контроль знань здобувачів вищої освіти передбачає оцінювання за наступними основними напрямками:

- перевірка теоретичних знань (у тому числі тем, винесених на самостійне вивчення);
- перевірка підготовки до лабораторних занять.

З даних компонентів складаються загальні бали, які фіксуються в журналі викладача.

Модульний контроль проводиться задля оцінки результатів навчання після закінчення логічно завершеної частини лекційних та лабораторних занять– змістового модуля.

Оцінювання рівня засвоєння теоретичних знань здобувачів вищої освіти проводиться під час усної співбесіди зі здобувачами по теоретичним матеріалам, за результатами захисту лабораторних робіт й виконання самостійних робіт. Підсумковим контролем є екзамен.

10. РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ ЗДОБУВАЧІ

Використовується модульно-рейтингова система оцінювання, яка передбачає розподіл балів за виконання всіх запланованих видів робіт. Для денної форми навчання ця сума складається з балів, отриманих за контрольню-модульну роботу, та балів, що їх накопичив студент у ході поточного контролю. При цьому максимальна кількість балів при умові бездоганного виконання дорівнює 60 балів за 1 змістовий модуль та 40 балів за 2 змістовий модуль.

Індивідуальна контрольна робота (ІКР), яку виконують студенти-заочники, може дати сумарно за два змістові модулі максимально 100 балів при найвищій якості виконання. При зниженні якості виконання контрольної роботи відповідно знижується і сума балів.

Шкала оцінювання лабораторних робіт

№ змістового модуля	Вид роботи	Тема	Кількість задач	Максимальна кількість балів		
				денна ф.н.	заочна ф.н.	
Змістовий модуль 1	Лабораторна робота № 1	Множини	3	10	10	
	Лабораторна робота № 2	Відношення та операції над множинами	3	10	10	
	Лабораторна робота № 3	Відношення між елементами множин	3	10	10	
	Лабораторна робота № 4	Булеві функції	3	10	10	
	Лабораторна робота № 5	Комбінаторика	3	10	20	
	<i>Разом за лабораторні роботи / індивідуальна КР</i>				50	60
	<i>Контрольно-модульна робота змістового модуля 1</i>				10	
	<i>Разом за змістовий модуль 1</i>				60	60
Змістовий модуль 2	Лабораторна робота № 6	Кодування	3	10	10	
	Лабораторна робота № 7	Графи	3	10	10	
	Лабораторна робота №8	Задачі на графах	3	10	20	
	<i>Разом за лабораторні роботи / індивідуальна КР</i>				30	40
	<i>Контрольно-модульна робота змістового модуля 2</i>				10	-
	<i>Разом за змістовий модуль 2</i>				40	40
<i>Разом за заліковий модуль</i>				100	100	

Кожна лабораторна робота містить по 3 завдання різного рівня складності. Виконання кожного завдання оцінюється у 2, 3 або 5 балів відповідно до його складності. Студент не зобов'язаний виконувати всі завдання. Таким чином досягається диференціація навчання з урахуванням освітніх потреб, компетентностей та можливостей кожного здобувача.

Загальне оцінювання лабораторної роботи ведеться за наступними показниками:

1. Своєчасність практичного виконання лабораторної роботи (у тиждень згідно із графіком робіт для студентів денної форми навчання або за 1 тиждень до сесії для студентів заочної форми навчання) (0-4 бали).

2. Теоретичний захист виконаної лабораторної роботи (у тиждень наступний за тижнем планового виконання роботи для студентів денної форми навчання або за 1 тиждень до сесії для студентів заочної форми навчання) (0-4 бали).

3. Якість знайдених студентом рішень (ефективність алгоритму, доречність використання елементів інтерфейсу тощо) (3-12 балів).

Якість знайдених студентом рішень оцінюється наступним чином:

1. Робота виконана у повному обсязі без зауважень - максимальний бал;
2. Робота виконана достатньо повно з деякими зауваженнями – 75% від максимального бала;
3. Робота виконана не повністю – 50% від максимального бала.

Модульний контроль здійснюється 2 рази за семестр на 9 та 18-ому тижнях четвертого семестру.

Контрольно-модульні роботи складаються з контрольних питань. Оцінювання *контрольних питань* розподіляється пропорційно їх кількості.

КМР відображує теоретичні знання і може дати максимально 10 балів при найвищій якості виконання. При зниженні якості КМР знижується і сума балів відповідно до шкали, що наводиться у таблиці:

Шкала оцінювання контрольних робіт

Відсоток вірних компонентів КМР	0 – 30	31 – 60	61 – 75	76 – 85	86 – 94	95 – 100
Сума балів за КМР	0	2	4	6	8	10

Для допуску до підсумкового контролю студент повинен виконати графік навчального процесу, усі види запланованих завдань і протягом семестру отримати в сумі не менше 50 балів.

Семестровий контроль здійснюється у формі екзамену в четвертому семестрі для всіх форм навчання. Екзаменаційний білет складається з 2-х теоретичних питань та 1-ї задачі, для виконання екзаменаційних завдань студенту надається 90 хвилин.

Результати екзамену оцінюються за 100-бальною шкалою. Підсумкова оцінка з дисципліни розраховується як середня зважена з оцінок за модулі та за екзамен, за формулою:

$$O = ((ОПК1+ОПК2) * 0,75 + ОЕ * 0,25), \text{ де}$$

O – підсумкова оцінка з дисципліни за семестр, бали;

ОПК1, ОПК2 – сумарна кількість балів, що отримав здобувач за змістові модулі;

ОЕ – кількість балів, що отримав здобувач під час складання екзамену в період екзаменаційної сесії.

У відомість оцінка проставляється як у балах національної шкали, так і за шкалою ECTS.

При наявності у здобувачів **результатів неформального навчання** за освітнім компонентом «Дискретні структури» у повному обсязі, визнання та оцінювання результатів здійснюється відповідно до «Положення про порядок визнання у Криворізькому національному університеті результатів навчання, отриманих в умовах неформальної освіти». У випадку, якщо за підсумками визнання результатів неформального навчання визнається тільки частина результатів навчання, заявнику зараховуються окремі види навчальної роботи за освітнім компонентом «Дискретні структури».

Нижче наведені окремі види навчальної роботи, які можуть бути зараховані здобувачеві при наявності сертифікату про успішне проходження рекомендованих онлайн курсів.

Тема	Посилання на рекомендовані курси
Множини	https://www.lektorium.tv/diskretnaya-matematika https://www.udemy.com/course/discrete-math/?couponCode=ST18MT12125CROW
Булеві функції	https://www.lektorium.tv/diskretnaya-matematika https://www.udemy.com/course/discrete-math/?couponCode=ST18MT12125CROW
Комбінаторика	https://www.lektorium.tv/diskretnaya-matematika https://www.coursera.org/learn/analytic-combinatorics
Графи	https://prometheus.org.ua/prometheus-free/algorithms-design-analysis-part1/ https://www.coursera.org/learn/trees-graphs-basics

Шкала оцінювання

Національна шкала успішності	Оцінка ECTS	Визначення ECTS	100-бальна система оцінювання
відмінно/ зараховано	A	ВІДМІННО - відмінне виконання лише з незначними помилками	90...100
добре/ зараховано	B	ДУЖЕ ДОБРЕ - вище середнього рівня з кількома помилками	80...89
	C	ДОБРЕ - у цілому правильно робота з певною кількістю помилок і недоліків	71...79
задовільно/ зараховано	D	ЗАДОВІЛЬНО - непогано, але зі значною кількістю грубих помилок	61...70
	E	ДОСТАТНЬО - виконання задовольняє мінімальні потреби	50...60
незадовільно/ не зараховано	FX	НЕЗАДОВІЛЬНО - із можливістю повторного складання	30...49
	F	НЕЗАДОВІЛЬНО - з обов'язковим повторним вивчення дисципліни	0...29

11. ПЕРЕЛІК ПИТАНЬ ДЛЯ ПІДСУМКОВОГО КОНТРОЛЮ ЗНАНЬ, УМІНЬ, НАВИЧОК

1. Множини. Визначення. Основні поняття.
2. Парадокс Рассела. Причини виникнення та можливості подолання.
3. Діаграми Ейлера та Ейлера-Венна.
4. Відношення між множинами.
5. Операції над множинами.
6. Відносини між множинами. Властивості відносин. Представлення відносин.
7. Алгебраїчні структури. Основні поняття.
8. Типи алгебраїчних структур.
9. Булеві функції з однією змінною.
10. Булеві функції з двома змінними.
11. Логічні вирази і обчислення.
12. Комбінаторика. Типові комбінаторні задачі.
13. Кодування. Представлення різних типів даних у двійковій формі.
14. Кодування. Теорема Шеннона про оптимальне кодування.
15. Кодування. Коди Хаффмана.
16. Кодування. Арифметичне кодування.
17. Кодування. Словникові алгоритми кодування.
18. Кодування. Алгоритми завадостійкого кодування.
19. Графи. Визначення. Основні поняття.
20. Представлення графів в ЕОМ.
21. Алгоритми пошуку на графах.
22. Ейлеровий та Гамільтонів цикли.
23. Пошук найкоротшого шляху. Алгоритм Дейкстри.
24. Пошук найкоротшого шляху. Алгоритм Флойда-Уоршила.
25. Потoki в мережах. Пошук максимального потоку.
26. Деревя. Визначення. Основні поняття.
27. Деревя. Представлення в ЕОМ.
28. Деревя. Побудова мінімального остового дерева. Алгоритм Крускала.
29. Деревя. Побудова мінімального остового дерева. Алгоритм Прима.
30. Задача розфарбування графа. Теорема про п'ять фарб.

Приклад екзаменаційного білету

1. Множини. Визначення. Основні поняття.
2. Кодування. Арифметичне кодування.
3. Існують множини $\{0,1,2\}$ та $\{2,3\}$. Їх декартовий добуток дорівнює:
 - a. $\{(0,2), (0,3), (1,2), (1,3), (2,2), (2,3)\}$
 - b. $\{0, 0, 2, 3, 4, 6\}$
 - c. $\{\{0,2\}, \{0,3\}, \{1,2\}, \{1,3\}, \{2,2\}, \{2,3\}\}$

12. НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ З ДИСЦИПЛІНИ

12.1 Навчальна та довідкова література

1. Комп'ютерна дискретна математика: навч. посіб. / А. М. Сергієнко, А. А. Молчанова, В. О. Романкевич. Київ: КПІ ім. Ігоря Сікорського, 2022. 189 с.
2. Дискретні структури (Алгебраїчні та числові системи, комбінаторний аналіз): навч.-метод. посіб. / Бойко І.В., Петрик М.Р., Цуприк Г.Б. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2017. 64 с.
3. Борисенко О. А. Дискретна математика. Київ: Університетська книга, 2023. 255 с.

12.2. Методична література

1. Стрюк А.М. Методичні вказівки до виконання лабораторних робіт з дисципліни «Дискретні структури» для студ. всіх форм навч. за спец. 121 «Інженерія програмного забезпечення». Кривий Ріг, 2024. 36 с.

13. ІНФОРМАЦІЙНІ РЕСУРСИ

До складу інформаційних ресурсів навчальної дисципліни входять:

1. Бібліотека Криворізького національного університету. URL: <http://lib.knu.edu.ua/>.

Internet-ресурси:

2. Стрюк А.М. Дискретні структури: консп. лекцій та презентації. URL: <https://classroom.google.com/c/NjUxNTEyNTE4NDIy?cjc=бенmsut> (дата звернення 08.01.2025).

14. ТЕРМІНОЛОГІЧНИЙ СЛОВНИК

Дискретна математика (Discrete mathematics) - is the study of mathematical structures that can be considered "discrete" (in a way analogous to discrete variables, having a bijection with the set of natural numbers) rather than "continuous" (analogously to continuous functions). Objects studied in discrete mathematics include integers, graphs, and statements in logic.

Множина (A Set) - is a collection of different things; these things are called elements or members of the set and are typically mathematical objects of any kind: numbers, symbols, points in space, lines, other geometrical shapes, variables, or even other sets. A set may have a finite number of elements or be an infinite set. There is a unique set with no elements, called the empty set; a set with a single element is a singleton.

Потужність множини (The cardinality of a set) - or the cardinal number of a set, is a characteristic of sets (including infinite ones), which generalizes the concept of the number (number) of elements of a finite set. This concept is based on natural ideas about the comparison of sets. Any two sets between the elements of which a one-to-one correspondence (bijection) can be established contain the same number of elements (have the same cardinality). Conversely, sets equal in cardinality must admit such a one-to-one correspondence. A part of a set does not exceed the complete set in cardinality (i.e., in the number of elements). Cardinality of sets allows us to compare infinite sets. For example, countable sets are the "smallest" infinite sets.

Діаграма Ейлера (Euler diagram) - is a diagrammatic means of representing sets and their relationships. They are particularly useful for explaining complex hierarchies and overlapping definitions. They are similar to another set diagramming technique, Venn diagrams. Unlike Venn diagrams, which show all possible relations between different sets, the Euler diagram shows only relevant relationships.

Діаграма Венна (Venn diagram) - is a widely used diagram style that shows the logical relation between sets, popularized by John Venn. The diagrams are used to teach elementary set theory, and to illustrate simple set relationships in probability, logic, statistics, linguistics and computer science. A Venn diagram uses simple closed curves drawn on a plane to represent sets. Very often, these curves are circles or ellipses.

Алгебра множин (the Algebra of sets) – is a branch of set theory that defines the laws of composition of sets based on the basic properties of operations on them, and also proposes a certain systematic procedure for computing set-theoretic equations and relations.

Алгебраїчна структура (an Algebraic structure) - in mathematics consists of a nonempty set A (called the underlying set, carrier set or domain), a collection of operations on A (typically binary operations such as addition and multiplication), and a finite set of identities (known as axioms) that these operations must satisfy. An algebraic structure may be based on other algebraic structures with operations and axioms involving several structures.

Гомоморфізм (A Homomorphism) – is a structure-preserving mapping between two algebraic structures of the same type (e.g., two groups, two rings, two vector spaces). Homomorphisms of two vector spaces are also called linear mappings, and their study is dealt with in linear algebra. The concept of homomorphism has been generalized under the name morphism for many structures that do not have a supporting set or are not algebraic. This generalization is the starting point of category theory. A homomorphism can also be an isomorphism, an endomorphism, an automorphism, etc. Each of these homomorphisms can be defined in a way that can be generalized to any class of morphisms.

Алгебра логіки (Булева алгебра) (Boolean algebra) - is a branch of mathematical logic that studies the system of logical operations on statements. In the algebra of logic, the values of variables are truth values, true or false, usually defined as 1 and 0, respectively. In contrast to elementary algebra, in which the values of variables are numbers and the basic operations are addition and multiplication, the basic operations of Boolean algebra are conjunction, disjunction, and negation. Thus, the formalism for describing logical relations is analogous to how numerical relations are described in elementary algebra.

Кон'юнкція (Conjunction) - (AND operation) is a binary operation that evaluates to "true" if all operands evaluate to "true." The operation involves the use of the conjunction "and" in logical expressions.

Диз'юнкція (Disjunction) - (OR operation) is a two-place logical operation that evaluates to "true" if at least one of the operands evaluates to "true".

Заперечення (Negation) - (NOT operation) in logic is a unary operation on judgments, the result of which is a judgment (in a certain sense) "opposite" to the original one.

Комбінаторика (Combinatorics (Combinatory analysis)) - is a branch of mathematics devoted to solving problems of selecting and arranging elements of some, usually finite, set according to given rules. Each such rule determines the method of constructing some construction from the elements of the initial set, which is called a combinatorial configuration. Therefore, the goal of combinatorial analysis is the study of combinatorial configurations, algorithms for their construction, optimization of such algorithms, and also the solution of enumeration problems.

Перестановка (A Permutation) - of a finite set is an ordered set of its elements without repetitions.

Розміщення (an Arrangement) - in combinatorics, an arrangement of n elements by m , or an ordered (n, m) selection from a set M (radix n , $m \leq n$) is an arbitrary tuple consisting of m pairwise distinct elements.

Поєднання (a Combination or compound) - in mathematics, is a way of selecting several things from a larger group, where the order is not important. In the case of small numbers, it is possible to count the number of compounds. More formally, a k -combination of a set S is the subset formed by k distinct elements of S . Compounds may or may not allow repetitions. The number of combinations with repetitions from n to k is equal to the number of combinations without repetitions from $(n+k-1)$ to k .

Теорія кодування (Coding theory) - is the study of the properties of codes and their respective fitness for specific applications. Codes are used for data compression, cryptography, error detection and correction, data transmission and data storage. Codes are studied by various scientific discipline such as information theory, electrical engineering, mathematics, linguistics, and computer science for the purpose of designing efficient and reliable data transmission methods. This typically involves the removal of redundancy and the correction or detection of errors in the transmitted data.

Вихідне кодування (Source coding) - is the process of encoding information using fewer bits than the original representation. Any particular compression is either lossy or lossless. Lossless compression reduces bits by identifying and eliminating statistical redundancy. No information is lost in lossless compression. Lossy compression reduces bits by removing unnecessary or less important information. Typically, a device that performs data compression is referred to as an encoder, and one that performs the reversal of the process (decompression) as a decoder.

Бит (Bit) - is the most basic unit of information in computing and digital communication. The name is a portmanteau of binary digit. The bit represents a logical state with one of two possible values. These values are most commonly represented as either "1" or "0", but other representations such as true/false, yes/no, on/off, or +/- are also widely used.

Алгоритм Гаффмана (a Huffman code) - is a particular type of optimal prefix code that is commonly used for lossless data compression. The output from Huffman's algorithm can be viewed as a variable-length code table for encoding a source symbol. The algorithm derives this table from the estimated probability or frequency of occurrence (weight) for each possible value of the source symbol. As in other entropy encoding methods, more common symbols are generally represented using fewer bits than less common symbols. Huffman's method can be efficiently implemented, finding a code in time linear to the number of input weights if these weights are sorted. However, although optimal among methods encoding symbols separately, Huffman coding is not always optimal among all compression methods – it is replaced with arithmetic coding or asymmetric numeral systems if a better compression ratio is required.

Алгоритм Шеннона-Фано (a Shannon-Fano coding) - is one of two related techniques for constructing a prefix code based on a set of symbols and their probabilities (estimated or measured). Shannon-Fano codes are suboptimal in the sense that they do not always achieve the lowest possible expected codeword length, as Huffman coding does. However, Shannon-Fano codes have an expected codeword length within 1 bit of optimal. Fano's method usually produces encoding with shorter expected lengths than Shannon's method. However, Shannon's method is easier to analyse theoretically.

Теорія графів (A graph theory) - in mathematics and computer science graph theory is the study of graphs, which are mathematical structures used to model pairwise relations between objects. A graph in this context is made up of vertices (also called nodes or points) which are connected by edges (also called arcs, links or lines). A distinction is made between undirected graphs, where edges link two vertices symmetrically, and directed graphs, where edges link two vertices asymmetrically. Graphs are one of the principal objects of study in discrete mathematics.

Граф (A graph) - In graph theory, a graph is a structure consisting of a set of objects, where some pairs of objects are "connected" in some sense. The objects are represented by abstractions called vertices (also called nodes or points), and each connected pair of vertices is called an edge (also called a link or line). Typically, a graph is represented diagrammatically as a set of points or circles for vertices, connected by lines or curves for edges. Graphs can be directed or undirected.

Орієнтований граф (a Directed graph) - in mathematics, and more specifically in graph theory, a directed graph (or digraph) is a graph that is made up of a set of vertices connected by directed edges, often called arcs.

Дерево (a Tree) - is a widely used abstract data type that represents a hierarchical tree structure with a set of connected nodes. Each node in the tree can be connected to many children (depending on the type of tree), but must be connected to exactly one parent, except for the root node, which has no parent (i.e., the root node as the top-most node in the tree hierarchy). These constraints mean there are no cycles or "loops" (no node can be its own ancestor), and also that each child can be treated like the root node of its own subtree, making recursion a useful technique for tree traversal. In contrast to linear data structures, many trees cannot be represented by relationships between neighboring nodes (parent and children nodes of a node under consideration, if they exist) in a single straight line (called edge or link between two adjacent nodes).

Бінарне дерево (Binary Tree) - is a tree data structure in which each node has at most two children, referred to as the left child and the right child. That is, it is a k-ary tree with $k = 2$. A recursive definition using set theory is that a binary tree is a tuple (L, S, R) , where L and R are binary trees or the empty set and S is a singleton set containing the root.

Алгоритм Дейкстри (Dijkstra's algorithm) - a graph algorithm invented by Dutch scientist Edsger Dijkstra in 1959. It finds the shortest paths from one vertex of a graph to the rest. The algorithm works only for graphs without edges of negative weight. The algorithm is widely used in programming.

Алгоритм Флойда-Воршалла (Floyd-Warshall's algorithm) - is an algorithm for finding shortest paths in a weighted graph with positive or negative edge weights (but no negative cycles). In a single execution of the algorithm, the lengths (sum weights) of the shortest paths between all pairs of vertices are found. Although it does not return details of the paths themselves, it is possible to reconstruct the paths using simple modifications of the algorithm. Variants of the algorithm can also be used to find the transitive closure of a relation R, or the widest paths between all pairs of vertices in a weighted graph.

Алгоритм Крускала (Kruskal's algorithm) - finds a minimum spanning forest of an undirected edge-weighted graph. If the graph is connected, it finds a minimum spanning tree. It is a greedy algorithm that in each step adds to the forest the lowest-weight edge that will not form a cycle. The key steps of the algorithm are sorting and the use of a disjoint-set data structure to detect cycles. Its running time is dominated by the time to sort all of the graph edges by their weight. A minimum spanning tree of a connected weighted graph is a connected subgraph, without cycles, for which the sum of the weights of all the edges in the subgraph is minimal. For a disconnected graph, a minimum spanning forest is composed of a minimum spanning tree for each connected component.

Алгоритм Прима (Prim's algorithm) - is a greedy algorithm that finds a minimum spanning tree for a weighted undirected graph. This means it finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized. The algorithm operates by building this tree one vertex at a time, from an arbitrary starting vertex, at each step adding the cheapest possible connection from the tree to another vertex.

**Додаток до робочої програми
Робочий план**

з дисципліни «Алгоритмізація обчислювальних процесів»
для здобувачів вищої освіти першого (бакалаврського) рівня вищої освіти
за спеціальністю 121 «Інженерія програмного забезпечення»

Вид навчальної роботи	Годин у семестрі/ кредитів	Розподіл годин по тижнях																		Вид підсумкового контролю
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	
Лекційні заняття	54	3 ПК	3 ПК	3 ПК	3 ПК	3 ПК	3 ПК	3 ПК	3 ПК	3 ПК	3 ПК	3 ПК	3 ПК	3 ПК	3 ПК	3 ПК	3 ПК	3 ПК	3 ПК	екзамен
Лабораторні заняття	36	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ЗМ	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ПК	2 ЗМ	
Самостійна робота	120	5	5	6	6	7	7	7	7	7	7	7	7	7	7	7	7	7	7	
Всього годин/ кредитів	210/7	10	10	11	11	12	12	12	12	12	12	12	12	12	12	11	12	12	12	

Позначки:

ПК - поточний контроль
ЗМ - складання змістових модулів