КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

БУДІВНИЦТВА І АРХІТЕКТУРИ

**БАКАЛАВР**

Кафедра. інформаційних технологій проектування

та прикладної математики

.

«ЗАТВЕРДЖУЮ»

«ЗАТВЕРДЖУЮ»

В.о. декана факультету автоматизації і інформаційних технологій

 \_\_\_\_\_\_\_/ Олександр ТЕРЕНТЬЄВ /

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2024 року

РОБОЧА ПРОГРАМА ОСВІТНЬОЇ КОМПОНЕНТИ

"Дисципліни обов`язкової компоненти"

**«ТЕОРІЯ АЛГОРИТМІВ»**

|  |  |
| --- | --- |
| шифр | назва спеціальності, освітньої програми |
| 123 | «Комп’ютерна інженерія»  |

|  |
| --- |
| Розробники: |
| Серпінська О.І., асистент |  |
| (прізвище та ініціали, науковий ступінь, звання) | (підпис) |
|  |  |
| (прізвище та ініціали, науковий ступінь, звання) | (підпис) |

Робоча програма затверджена на засіданні кафедри \_\_\_\_\_\_\_\_\_\_\_\_\_

протокол № від « » 2024 року

Завідувач кафедри /Олександр ТЕРЕНТЬЄВ/

 (підпис)

Схвалено гарантом освітньої програми \_\_\_\_\_\_\_\_\_\_\_

Гарант ОП /Євгенія ШАБАЛА/

 (підпис)

Розглянуто на засіданні науково-методичної комісії спеціальності

протокол № від « » 2024 року

**ВИТЯГ З РОБОЧОГО НАВЧАЛЬНОГО ПЛАНУ**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| шифр |  | Форма навчання: **денна** | Форма контролю | Семестр | Відмітка про погодження заступником декана факультету |
| Назва спеціальності, освітньої програми | Кредитів на сем. | Обсяг годин | Кількість індивідуальних робіт |
| Всього | аудиторних | Сам.роб. |
| Разом | у тому числі |
| Л | Лр | Пз | КП | КР | РГР | Конт.роб |
| 123 | Комп`ютерна інженерія (в т.ч. скорочений термін навчання) | ***4,0*** | ***120*** | ***60*** | ***20*** | ***30*** |  | ***70*** |  |  | 1 |  | ***залік*** |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| шифр |  | Форма навчання: **заочна (вечірня)** | Форма контролю | Семестр | Відмітка про погодження заступником декана факультету |
| Назва спеціальності, освітньої програми | Кредитів на сем. | Обсяг годин | Кількість індивідуальних робіт |
| Всього | аудиторних | Сам.роб. |
| Разом | у тому числі |
| Л | Лр | Пз | КП | КР | РГР | Конт.роб |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Мета та завдання освітньої компоненти**

Мета дисципліни:

Отримання студентами ґрунтовної математичної підготовки та знань теоретичних, методичних і алгоритмічних основ інформаційних технологій для їх використання під час розв'язання прикладних і наукових завдань в області інформаційних систем і технологій, забезпечення теоретичної та інженерної підготовки фахівців у галузі проектування, впровадження та використання інформаційних систем в бізнесі.

**Завданням** вивчення дисципліни "Теорія алгоритмів" є теоретична та практична підготовка майбутніх фахівців з таких питань:

* Математичні основи аналізу алгоритмів.
* Алгоритмічні стратегії.
* Основи теорії обчислюваності.
* Класи складності Р і NP.
* Алгоритми сортування, злиття та пошуку.
* Комбінаторні алгоритми.
* Рекурсивні алгоритми.
* Фундаментальні алгоритми на графах і деревах.

**Компетентності здобувачів освітньої програми, що формуються в результаті засвоєння освітньої компоненти**

|  |  |
| --- | --- |
| **Код** | **Зміст компетентності** |
| **Інтегральна компетентність** |
| **ІК** | Здатність розв’язувати складні спеціалізовані задачі та практичні проблеми під час професійної діяльності в комп’ютерній галузі або навчання, що передбачає застосування теорій та методівкомп’ютерної інженерії і характеризується комплексністю та невизначеністю умов. |
| **Загальні компетентності** |
| Z 1 | Здатність до абстрактного мислення, аналізу і синтезу. |
| Z 2 | Здатність вчитися і оволодівати сучасними знаннями. |
| Z 3 | Здатність застосовувати знання у практичнихситуаціях. |
| Z 7 | Вміння виявляти, ставити та вирішувати проблеми. |
| **Фахові компетентності** |
| Р 2 | Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення. |
| Р 7 | Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп’ютернихсистем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності. |
| Р 11 | Здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів. |
| Р 15 | Здатність аргументувати вибір методів розв’язування спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати тазахищати прийняті рішення. |

**Програмні результати здобувачів освітньої програми, що формуються в результаті засвоєння освітньої компоненти**

|  |  |
| --- | --- |
| **Код** | **Програмні результати** |
| N 1 |  Знати і розуміти наукові положення, що лежать в основі функціонування комп’ютерних засобів, систем та мереж. |
| N 2 | Мати навички проведення експериментів, збирання даних та моделювання в комп’ютерних системах. |
| N 6 | Вміти застосовувати знання для ідентифікації, формулювання і розв’язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей. |
| N 8 | Вміти системно мислити та застосовувати творчіздібності до формування нових ідей. |
| N 14 | Вміти поєднувати теорію і практику, а також приймати рішення та виробляти стратегію діяльності для вирішення завдань спеціальності з урахуванням загальнолюдських цінностей, суспільних, державнихта виробничих інтересів. |
| N 16 | Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення. |

**Програма дисципліни**

**Змістовий модуль 1. Аналіз алгоритмів та алгоритмічні стратегії**

***Лекція 1*** Основи аналізу алгоритмів

**Тема 1.** Зміст дисципліни "Теорія алгоритмів", її зв’язок із іншими дисциплінами.​

**Тема 2.** Поняття алгоритму. Способи задання.​

**Тема 3.** Властивості алгоритмів

**Тема 4.** Етапи проектування і аналізу алгоритмів

**Тема 5.** Основи аналізу ефективності алгоритмів​

Тема 6. Ефективність алгоритму в різних випадках

Висновки

- ознайомлення слухачів з основами теорії алгоритмів, з методами оцінки складності алгоритмів і розробки ефективних алгоритмів.​

- формалізація поняття «алгоритм» і дослідження формальних алгоритмічних систем;​

***Лабораторне заняття 1.*** Алгоритми. Основні поняття та властивості.

Зміст заняття:

1. Скласти алгоритм лінійної структури.
2. Скласти алгоритм розгалуженої структури.
3. Скласти алгоритм циклічної структури.
4. Скласти алгоритм з використанням ітераційного циклу.

***Лекція 2*** Структури даних

**Тема 1.** Проста змінна.

**Тема 2.** Масив

**Тема3.** Стек.

**Тема 4.** Черга.

Висновки

знання різноманітних структур даних, ефективне їх використання мають неабияке

значення. Правильне, коректне визначення структур даних, що використовуються в

алгоритмі, - це значна частина успіху.​

Можна сказати, що структури даних є одним з інструментів розробки алгоритмів. Багато відомих методів алгоритмів базуються на тих чи інших структурах.​

***Лабораторне заняття 2.*** Стеки. Реалізація стеків за допомогою масивів та покажчиків.

Зміст заняття:

1. Написати програму реалізації стеків за допомогою масивів і покажчиків.
2. Написати програму реалізації стеків, використовуючи “Масив вгору”.

Виконати: додавання елемента в стек; видалення елемента зі стеку; перевірка, чи порожній стек; перегляд елемента у вершині стека без видалення; очищення стека.

***Лекція 3.***Асимптотичний аналіз. Оцінки складності алгоритмів​

**Тема 1.** Основи асимптотичного аналізу​

**Тема 2.** Оцінки складності алгоритмів (О, Ώ, Θ )

**Тема 3.** Використання границь для порівняння порядку росту двох функцій​

Висновки. Асимптотичний аналіз  математично оцінює час виконання алгоритму підрахунком операцій. ​

Метою асимптотичного аналізу є порівняння витрат часу та інших ресурсів різними алгоритмами, призначеними для рішення однієї і тієї ж задачі, при великих обсягах вхідних даних.​

Оцінка функції ефективності, що використовується в асимптотичному аналізі, називається складністю алгоритму, і дозволяє визначити, як швидко росте трудомісткість алгоритму зі збільшенням обсягу даних. ​

***Лабораторне заняття*** **3**. Черги. Реалізація черг за допомогою циклічних масивів та покажчиків.

Зміст заняття:

1. Написати програму реалізації черги за допомогою покажчиків.
2. Написати програму реалізації черги за допомогою циклічного масиву.

Виконати: додавання елемента в чергу; видалення елемента з черги; перевірка, чи порожня черга; перегляд елемента в голові черги; очищення черги.

***Лекція 4.*** Найпростіша модель обчислювання.

**Тема 1.** Машина Тьюринга.

**Тема 2.** Нормальний алгоритм Маркова

**Тема 3.** Блок-схема Поста

Висновки.

- Якщо задача має рішення, то відомий хоча б один алгоритм її розв'язання.​

- Якщо задачу вирішити не можна, то її слід віднести до розряду алгоритмічно нерозв'язних.​

***Лабораторне заняття*** **4.** Машина Тьюринга. Нормальний алгоритм Маркова.

Зміст заняття:

1. Скласти програму роботи для Машини Тьюринга, де *Р* – вхідне непорожнє слово (length(*P*)>2), *А* – алфавіт вхідного слова, тобто набір символів з яких може складатися слово *Р.*
2. Скласти нормальний алгоритм Маркова, де *Р* – вхідне непорожнє слово (length(*P*)>2), *А* – алфавіт вхідного слова, тобто набір символів з яких може складатися.

***Лекція 5.*** Алгоритми внутрішнього сортування.

Тема 1. Сортування обміном (бульбашкою).

Тема 2. Сортування вибором.

Тема 3. Сортування вставками.

Тема 4. Сортування методом Шелла.

Тема 5. Швидке сортування (метод Хоара, Quick Sort).

Тема 6. Алгоритм пірамідального сортування.

Висновки. Ефективність алгоритму залежить від наступних факторів:​

- скільки елементів необхідно відсортувати;​

- в якій степені елементи вже відсортовані;​

- який діапазон і розподілення значень елементів, що сортуються;​

- записані елементи в файл або в масив, тощо.​

***Лабораторне заняття*** **5.** Алгоритми внутрішнього сортування.

Зміст заняття:

1. Відсортувати масив методами внутрішнього сортування: бульбашкою, вставками, вибором, методом швидкого сортування Хоара, методом Шелла.
2. Порівняти алгоритми за складністю.
3. Скласти блок-схеми алгоритмів методів.
4. Написати програми реалізації методів внутрішнього сортування за варіантами.

***Лекція 6.*** Алгоритми пошуку підрядків в рядках.

**Тема 1.** Постановка задачі.

**Тема 2.** Алгоритм послідовного (прямого) доступу

**Тема 3.** Алгоритм Рабіна -\_Карпа.

**Тема 4.** Алгоритм Кнута - Морриса - Пратта.

Висновки.

- розглянуто основні алгоритми пошуку підрядків в рядках;

- проведений порівняльний аналіз роботи алгоритмів.

***Лабораторне заняття*** **6.** Алгоритми пошуку підрядків в рядках.

Зміст заняття:

1. Задано: текст T, підрядок P, який треба знайти, просте число q (згідно варіанту), алфавіт , *d*=10, 
2. Знайти всі входження зразка P в текст Т, використовуючи алгоритм Рабіна-Карпа.
3. Знайти всі входження зразка P в текст Т, використовуючи алгоритм Кнута-Морриса-Пратта.
4. Скласти алгоритми та написати програму реалізації алгоритмів РК і КМП за варіантами.
5. Провести порівняльний аналіз алгоритмів пошуку в рядках.

***Лекція 7.*** Дерева. Основні операції з деревами. Алгоритми пошуку остовних дерев.

Тема 1. Кореневі дерева. Основні поняття.

Тема 2. Бінарні дерева пошуку. Алгоритм вставки елемента.

Тема 3. Бінарні дерева пошуку. Алгоритм видалення елемента.

Тема 4. Бінарні дерева пошуку. Алгоритм пошуку елемента.

Тема 5. Обхід дерев.

Тема 6. Дерева виразів.

Тема 7. Алгоритм DFS пошука в глибину.

Тема 8. Алгоритм ВFS пошука в ширину.

Тема 9. Алгоритм пошуку максимальної кількості остовних дерев.

Висновки.

- розглянуто основні алгоритми виконання операцій в бінарних деревах пошуку;

- розглянуто всі можливості обходу бінарного дерева;

- надано поняття дерева виразів;

- розглянуто алгоритми пошуку і пошуку максимальної кількості остовних дерев.

***Лабораторне заняття*** **7.** Бінарні дерева.

Зміст заняття:

1. Створити бінарне дерево пошуку (для кожного вузла дерева у всіх лівих вузлах повинні бути числа менші, а в правих - більші, ніж числа, що зберігаються в цьому вузлі), елементи якого вводяться з клавіатури й мають цілий тип.
2. Здійснити обхід дерева в прямому, зворотньому і симетричному порядку.
3. Написати функцію, яка знаходить найбільший і найменший елементи дерева.
4. Написати процедуру, яка видаляє з дерева всі парні елементи.
5. Написати процедуру, яка визначає число входжень заданого елемента в дерево.
6. Написати функцію, яка підраховує суму елементів дерева.

***Лекція 8.*** Алгоритми кодування інформації за допомогою дерев. Остовні дерева мінімальної ваги.

Тема 1. Стиснення даних.

Тема 2. Кодування Хаффмана.

Тема 3. Алгоритм Шеннона-Фано.

Тема 4. Алгоритм Прима.

Тема 5. Алгоритм Крускала.

Висновки.

- розглянуто остовні дерева мінімальної ваги;

- розглянуто основні принципи і алгоритми стиснення і кодування інформації за допомогою дерев.

***Лабораторне заняття 8*.** Алгоритми знаходження остовного дерева мінімальної ваги.

Зміст заняття:

1. Для заданого (за варіантом) графа побудувати остовне дерево мінімальної ваги за допомогою алгоритму Прима.
2. Для заданого (за варіантом) графа побудувати остовне дерево мінімальної ваги за допомогою алгоритму Крускала.
3. Скласти схеми алгоритмів і написати програми, що реалізують ці алгоритми.
4. Написати процедуру обчислення вартості побудованого остовного дерева.

***Лекція 9.*** Алгоритми пошуку найкоротшого шляху в графі. Алгоритми пошуку з поверненням.

Тема 1. Алгоритм Дейкстри.

Тема 2. Алгоритм Флойда.

Тема 3. Знаходження гамільтонових циклів в графі.

Тема 4. Переборний алгоритм розфарбовування вершин графа.

Висновки.

- Задача про найкоротший шлях полягає у знаходженні найкоротшого шляху від заданої початкової вершини а до заданої вершини z. Алгоритми Дейкстри і Флойда реалізують задачі, що є безпосередніми узагальненнями сформульованої задачі про найкоротший шлях.

- У разі, якщо побудова розв'язку є складною процедурою, то фактично під час роботи будується дерево можливих кроків алгоритму, а потім - у разі невдачі - відрізаються відповідні гілки дерева, доки не буде побудовано той шлях, що веде до успіху. Проходження вздовж гілок дерева та відходження у разі невдач і є алгоритм з поверненнями.​

***Лекція 10.*** Алгоритми зовнішнього сортування.

Тема 1. Пряме злиття.

Тема 2. Природне злиття.

Тема 3. Збалансоване багатоканальне злиття.

Тема 4. Багатофазне сортування.

Висновки.

-Зовнішнім сортуванням називають сортування послідовних файлів, які розташовані в зовнішній пам’яті і занадто великі, для того, щоб можливо було повністю перемістити їх в основну пам’ять і застосувати один з методів внутрішнього сортування. ​

-Найчастіше зовнішнє сортування використовується в системах управління базами даних при виконанні запитів, і від ефективності методів, які застосовуються, суттєво залежить продуктивність СУБД.

**Змістовий модуль 2. Розрахунково-графічна робота.**

**Тематика і зміст розрахунково-графічної роботи**

Тематика і зміст розрахунково-графічної роботи обумовлені основними розділами робочої навчальної програми дисципліни «Теорія алгоритмів» і орієнтовані на практичне застосування вивчених алгоритмів для конкретних задач та розробку програми їх реалізації зі складними структурами даних.

Конкретну тему розрахунково-графічної роботи студент обирає самостійно чи за вказівкою викладача з урахуванням рівня підготовки студента. У разі практичної участі студента в науково-дослідних роботах, що ведуться на кафедрі чи в університеті і мають безпосереднє відношення до дисципліни, студент може запропонувати власне формулювання теми роботи, узгодивши її з викладачем.

Тема та опис вхідних даних для розробки застосування вказуються в завданні на розрахунково-графічну роботу.

**Вимоги до розрахунково-графічної роботи**

Враховуючи основну ідею виконання розрахунково-графічної роботи «побудова алгоритму розв’язку задачі і лише потім написання програмного коду на мові програмування високого рівня», спочатку необхідно детально описати алгоритм розв’язку задачі словесним способом та представити графічним способом, за допомогою схеми алгоритму. Словесний спосіб описує послідовність кроків алгоритму в довільному викладенні на природній мові.

Для виконання роботи може бути використане будь-яке з популярних середовищ програмування.

Всі застосування повинні працювати в діалоговому режимі роботи і підтримувати дружній інтерфейс з користувачем.

Розробка програм повинна виконуватися з урахуванням вимог до мінімізації об’єму інформації, що зберігається у вивчених в теоретичному курсі структурах даних, типу об’єктів, списків, бінарних дерев тощо.

Тестові дані можуть формуватися безпосередньо в програмі чи завантажуватися з текстових файлів.

**Структура пояснювальної записки**

Пояснювальна записка повинна містити:

1. Титульний лист.

2. Завдання до розрахунково-графічної роботи.

3. Вступ.

4. Опис постановки задачі, результати аналізу задачі, вхідні та вихідні дані, обґрунтування вибору структур даних.

5. Опис розв’язку задачі словесним способом, використовуючи математичний апарат.

6. Схему алгоритму розв’язку задачі.

7. Оцінку коректності алгоритму.

8. Аналіз алгоритму на часову та просторову ефективності.

9. Опис програми (обґрунтування вибору мови програмування, опис основних процедур та функцій).

10. Контрольний приклад.

11. Список використаної літератури.

12. Додатки.

В додатках наводяться роздруківки текстів програмних модулів, растрових зображень основних екранних форм, тестових даних, результатів виконання застосування на тестових даних.

Пояснювальна записка оформлюється на листах формату А4, всі листи нумеруються та скріплюються.

**Оформлення пояснювальної записки**

До захисту роботу подають у вигляді спеціально підготовленого рукопису в прошитому вигляді.

Роботу друкують за допомогою комп’ютера на одній стороні аркуша білого паперу формату А4 (210х297 мм), дотримуючись таких вимог: Шрифт Times New Roman, Розмір шрифту 14 пунктів, Відстань між рядками 1,5 інтервали, Параметри сторінки Формат А4, Розташування Книжне.

Розміри поля: верхнє та нижнє – 20 мм, ліве – 20 мм, праве – 15 мм.

Абзацний відступ повинен бути однаковим впродовж усього тексту та дорівнювати п‘яти знакам (1,27 см). Формули та умовні знаки повинні бути введені до тексту за допомогою редакторів формул Microsoft Equation, Myth Type і т.п.

Ілюстрації (креслення, рисунки, графіки, схеми, діаграми, фотознімки) слід розміщувати в пояснювальній записці безпосередньо після тексту, де вони згадуються вперше, або на наступній сторінці. На всі ілюстрації мають бути посилання в пояснювальній записці. Посилання на ілюстрації роботи вказують порядковим номером ілюстрації, наприклад, «рис. 1.2».

Ілюстрації можуть мати назву, яку розміщують під ілюстрацією. За необхідності під ілюстрацією розміщують пояснювальні дані (підрисунковий текст). Ілюстрація позначається словом ”Рисунок \_\_\_ ”, яке разом з назвою ілюстрації розміщують після пояснювальних даних, наприклад ”Рисунок 3.1 Схема розміщення”.

На всі таблиці повинні бути посилання у тексті. Таблицю розміщують після першого згадування про неї в тексті, таким чином, щоб її можна було читати без повороту переплетеного блоку роботи або з поворотом за годинниковою стрілкою. Таблиця відокремлюється від тексту вільним рядком. Після назви таблиці вільний рядок не залишається.

Таблиці нумерують послідовно в межах розділів (таблиця 2.1 – перша таблиця другого розділу). Назва таблиці має бути стислою і відображати зміст.

Таблиця\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

назва таблиці

**Перелік завдань до розрахунково-графічної роботи**

1. Алгоритми на графах. Дводольний граф.

2. Алгоритми на графах. Розфарбування графа

3. Алгоритми на графах. Ізоморфний граф

4. Алгоритми на графах. Гамільтонів граф.

5. Алгоритми стискання даних. Код Хаффмана.

6. Алгоритми сортування Топологічне сортування.

7. Алгоритми на графах. Ейлерів граф.

8. Гра Hi-Q.

9. Триоміно.

10. Алгоритми на графах. Алгоритм Джонсона.

11. Алгоритми пошуку в рядках. Алгоритм Бойєра-Мура.

12. Головоломка «8».

13. Алгоритми пошуку в рядках. Алгоритм Хорспула.

14. Алгоритми на графах. Хвильовий алгоритм.

15. Алгоритми на графах. Алгоритм Форда-Фалкерсона.

16. Жадібні алгоритми. Задача про школу .

17. Алгоритм Джонсона.

18. Генерація лабіринту.

19. Алгоритми зовнішнього сортування.

Відсортувати файл, використовуючи багатофазне та природнє сортування.

20. Алгоритми зовнішнього сортування.

Відсортувати файл, використовуючи багатоканальне та пряме злиття.

**Методи контролю та оцінювання знань**

Загальне оцінювання здійснюється через вимірювання результатів навчання у формі пpoмiжнoгo (модульного) та підсумкового контролю (залік, захист розрахунково-графічної роботи тощо) відповідно до вимог зовнішньої та внутрішньої системи забезпечення якості вищої освіти.

**Політика щодо академічної доброчесності**

Тексти індивідуальних завдань (в т.ч. у разі, коли вони виконуються у формі презентацій або в інших формах) можуть перевірятись на плагіат. Для цілей захисту індивідуального завдання оригінальність тексту має складати не менше 70%. Виключення становлять випадки зарахування публікацій Здобувачів у матеріалах наукових конференціях та інших наукових збірниках, які вже пройшли перевірку на плагіат.

Списування під час тестування та інших опитувань, які проводяться у письмовій формі, заборонені (в т.ч. із використанням мобільних девайсів). У разі виявлення фактів списування з боку здобувача він отримує інше завдання. У разі повторного виявлення призначається додаткове заняття для проходження тестування.

**Політика щодо відвідування**

Здобувач, який пропустив аудиторне заняття з поважних причин, має продемонструвати викладачу та надати до деканату факультету документ, який засвідчує ці причини.

За об’єктивних причин (хвороба, міжнародне стажування, наукова та науково-практична конференція (круглий стіл) тощо) навчання може відбуватись в он-лайн формі за погодженням із керівником курсу.

**Методи контролю**

Основні форми участі Здобувачів у навчальному процесі, що підлягають поточному контролю: виступ на практичних заняттях; доповнення, опонування до виступу, рецензія на виступ; участь у дискусіях; аналіз першоджерел; письмові завдання (тестові, індивідуальні роботи у формі рефератів); та інші письмові роботи, оформлені відповідно до вимог. Кожна тема курсу, що винесена на лекційні та практичні заняття, відпрацьовується Здобувачами у тій чи іншій формі, наведеній вище. Обов’язкова присутність на лекційних заняттях, активність впродовж семестру, відвідування/відпрацювання усіх аудиторних занять, виконання інших видів робіт, передбачених навчальним планом з цієї дисципліни.

При оцінюванні рівня знань Здобувача аналізу підлягають:

- характеристики відповіді: цілісність, повнота, логічність, обґрунтованість, правильність;

- якість знань (ступінь засвоєння фактичного матеріалу): осмисленість, глибина, гнучкість, дієвість, системність, узагальненість, міцність;

- ступінь сформованості уміння поєднувати теорію і практику під час розгляду ситуацій, практичних завдань;

- рівень володіння розумовими операціями: вміння аналізувати, синтезувати, порівнювати, абстрагувати, узагальнювати, робити висновки з проблем, що розглядаються;

- досвід творчої діяльності: уміння виявляти проблеми, розв’язувати їх, формувати гіпотези;

- самостійна робота: робота з навчально-методичною, науковою, допоміжною вітчизняною та зарубіжною літературою з питань, що розглядаються, уміння отримувати інформацію з різноманітних джерел (традиційних; спеціальних періодичних видань, ЗМІ, Internet тощо).

**Тестове опитування** може проводитись за одним або кількома змістовими модулями. В останньому випадку бали, які нараховуються Здобувачу за відповіді на тестові питання, поділяються між змістовими модулями.

**Розрахунково-графічна робота** підлягає захисту Здобувачем на заняттях, які призначаються додатково.

Література, що рекомендується для виконання індивідуального завдання, наведена у цій робочій програмі, а в електронному вигляді вона розміщена на Освітньому сайті КНУБА, на сторінці кафедри.

Також як виконання індивідуального завдання за рішенням викладача може бути зарахована участь Здобувача у міжнародній або всеукраїнській науково-практичній конференції з публікацією у матеріалах конференції тез виступу (доповіді) на одну з тем, дотичних до змісту дисципліни, або публікація статті на одну з таких тем в інших наукових виданнях.

Текст індивідуального завдання подається викладачу не пізніше, ніж за 2 тижні до початку залікової сесії. Викладач має право вимагати від Здобувача доопрацювання індивідуального завдання, якщо воно не відповідає встановленим вимогам.

Результати поточного контролю заносяться до журналу обліку роботи. Позитивна оцінка поточної успішності Здобувачів за відсутності пропущених та невідпрацьованих практичних занять та позитивні оцінки за індивідуальну роботу є підставою для допуску до підсумкової форми контролю. Бали за аудиторну роботу відпрацьовуються у разі пропусків.

**Підсумковий контроль** здійснюється під час проведення залікової сесії з урахуванням підсумків поточного та модульного контроля. Під час семестрового контролю враховуються результати здачі усіх видів навчальної роботи згідно зі структурою кредитів.

Оцінювання проводиться за 100-бальною шкалою.

**Розподіл балів для дисципліни з формою контролю залік**

|  |  |  |
| --- | --- | --- |
| Поточне оцінювання | Залік  | Сума балів  |
| Змістові модулі | Інд. робота |
| 1 | 2 |
| 20 | 20 | 30 | 30 | 100 |

**Шкала оцінювання індивідуальної роботи**

|  |  |  |
| --- | --- | --- |
| **Оцінка за національною шкалою** | **Кількість** **балів** | **Критерії** |
| **відмінно** | 30 | відмінне виконання (розкриття теми, посилання та цитування сучасних наукових джерел (не старше 2017 року), **дотримання норм доброчесності**) |
| 25 | відмінне виконання з незначною кількістю помилок виконання (розкриття теми, посилання та цитування сучасних наукових джерел (більшість з яких не старше 2017 року), **дотримання норм доброчесності**) |
| **добре** | 22 | виконання вище середнього рівня з кількома помилками (розкриття теми в межах об`єкту та завдань роботи, посилання та цитування сучасних наукових джерел (серед яких є такі, що не старше 2017 року), **дотримання норм доброчесності**) |
| 20 | виконання з певною кількістю помилок (розкриття теми в межах об`єкту та завдань роботи, наявність посилань та цитувань наукових джерел, **дотримання норм доброчесності**) |
| **задовільно** | 18 | виконання роботи задовольняє мінімальним критеріям помилок (розкриття теми в основному в межах об`єкту роботи, наявність концептуального апарату роботи, присутність не менше 5 посилань та цитувань наукових джерел, **дотримання норм доброчесності**) |

**Шкала оцінювання: національна та ECTS**

|  |  |  |
| --- | --- | --- |
| Сума балів за всі види навчальної діяльності | ОцінкаECTS | Оцінка за національною шкалою |
| 90 – 100 | **А** | Зараховано |
| 82-89 | **В** |  |
| 74-81 | **С** |
| 64-73 | **D** |
| 60-63 | **Е**  |
| 35-59 | **FX** | Не зараховано з можливістю повторного складання |
| 0-34 | **F** | Не зараховано з обов’язковим повторним вивченням дисципліни |

**Умови допуску до підсумкового контроля**

Здобувачу, який має підсумкову оцінку за дисципліну від 35 до 59 балів, призначається додаткова залікова сесія. В цьому разі він повинен виконати додаткові завдання, визначені викладачем.

Здобувач, який не виконав вимог робочої програми по змістових модулях, не допускається до складання підсумкового контролю. В цьому разі він повинен виконати визначене викладачем додаткове завдання по змісту відповідних змістових модулів в період між основною та додатковою сесіями.

Здобувач має право на опротестування результатів контроля (апеляцію). Правила подання та розгляду апеляції визначені внутрішніми документами КНУБА, які розміщені на сайті КНУБА та зміст яких доводиться Здобувачам до початку вивчення дисципліни.

**Методичне забезпечення дисципліни**

**Підручники:**

1. Нікольський Ю. В., Пасічник В. В., Щербина Ю. М. Дискретна математика: підручник. – Львів: Магнолія-2006, 2010.- 431с.
2. Кормен Т., Лейзерсон Ч., Рівест Р. Алгоритми доступно: підручник. – К.: К.І.С, 2023. - 194с.
3. Угрин Д.І., УшенкоЮ.О., Ковальчук М.Л. Структури даних та алгоритми. Підручник. – Чернівці: Чернівецький національний університет ім. Ю. Федьковича, 2022. – 357 с.

**Навчальні посібники:**

1. Заяць В. М. Методи, алгоритми та програмні засоби для моделювання і аналізу динаміки складних об’єктів і систем на основі дискретних моделей: Монографія. – Львів: Новий Світ-2000, 2009. – 399с.
2. Матвієнко М.П. Теорія алгоритмів. Навчальний посібник. – К.: Видавництво Ліра-К, 2022. - 340 с.
3. Бартків А.Б., Гринчишин А.Т., Ломакович А.М. та ін. TURBO PASCAL: Алгоритми і програми: Числені методи в фізиці та математиці: Навч. посібник для студ. пед. ін-ів.- К.: Вища школа, 1992. – 248 с.
4. Блохін Л. М., Буриченко М.Ю., Кривоносенко О.П., Безкоровайний Ю.М. Базові алгоритми статистичної динаміки: Навч. посібник для студ. вищ. навч. закл. – К.: Книжкове вид-во НАУ, 2007. – 18 с.
5. Бех О. В., Городня Т. А., Щербак А. Ф. Збірник задач з математичного програмування: Навч. посібник для студ. вищ. навч. закл. - Львів: Магнолія, 2007.- 211 с.

**Конспекти лекцій:**

**Методичні роботи:**

1. Теорія алгоритмів: Методичні вказівки до виконання курсових робіт / Уклад. С.В. Білощицька, Н.Д. Федоренко, О.І. Баліна, І.С. Безклубенко, О.В. Доля, А.О. Білощицький. – К.: КНУБА, 2018. – с.

**Інформаційні ресурси:** http://org2.knuba.edu.ua/course/view.php?id=893