



АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ. ЧАСТИНА 2.

СТРУКТУРИ ДАНИХ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни	
Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	F Інформаційні технології
Спеціальність	F2 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці
Статус дисципліни	Нормативна
Форма навчання	Очна (денна)
Рік підготовки, семестр	1 курс, весняний семестр
Обсяг дисципліни	4 кредити, 120 годин, з яких 74 години аудиторних (30 год. лекції, 30 год. Лабораторні, 14 год. практичні), 46 годин становить самостійна робота
Семестровий контроль/ контрольні заходи	Залік/модульна контрольна робота.
Розклад занять	http://rozklad.kpi.ua/
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: д.т.н., проф. Барабаш Олег Володимирович, bar64@ukr.net Лабораторні заняття: Оленева Ксенія Миколаївна, oleneva_k@iit.kpi.ua Практичні заняття: д.т.н., проф. Барабаш Олег Володимирович, bar64@ukr.net
Розміщення курсу	Кампус
Програма навчальної дисципліни	

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Навчальна дисципліна «Алгоритми та структури даних. Частина 2. Структури даних» є тією частиною знань з програмування, яка пов'язана з умінням вирішувати завдання шляхом розробки алгоритмів та створення програм, які реалізують ці алгоритми. За останні десятиліття значно зросла роль створення програмних систем у сучасному світі. Це зумовлено потребою розв'язання важливих практичних задач в галузі інформаційних технологій.

Метою навчальної дисципліни є формування у студентів алгоритмічного мислення та набуття навиків розробки алгоритмів та програм на основі застосування різних структур даних для розв'язування теоретичних і практичних задач у професійній діяльності, а також розвитку

логічного та алгоритмічного мислення при виявленні та дослідженні закономірностей, яким підпорядковуються реальні процеси в сфері програмування.

Предметом вивчення є основи алгоритмізації та програмування на алгоритмічній мові, роботи в інтегрованих середовищах програмування, методики розробки програм необхідних для розв'язання професійних задач.

Програмні результати

Результатом вивчення навчальної дисципліни є формування у студентів **компетентностей**: — здатність до абстрактного мислення, аналізу та синтезу (ЗК 01);

— здатність застосовувати знання у практичних ситуаціях (ЗК 02);

— здатність до пошуку, оброблення та аналізу інформації з різних джерел (ЗК 06);

— володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних (ФК 07);

— здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення (ФК 08);

— здатність до алгоритмічного та логічного мислення (ФК 14);

— володіння алгоритмічним мисленням, методами програмної інженерії для реалізації програмного забезпечення з урахуванням вимог до його якості, надійності, виробничих характеристик.

Згідно з вимогами програми навчальної дисципліни студенти після вивчення дисципліни мають продемонструвати такі результати навчання:

— аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки (ПРН 01);

— вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання (ПРН 11);

— знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань (ПРН 13).

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Згідно з робочим навчальним планом навчальна дисципліна «Алгоритми та структури даних2. Структури даних» викладається студентам першого року підготовки першого (бакалаврського) рівня вищої освіти ступеня “бакалавр” спеціальності “Інженерія програмного забезпечення” освітньої програми “Інженерія програмного забезпечення інтелектуальних кіберфізичних систем в енергетиці” у першому навчальному семестрі. Дисципліна «Алгоритми та структури даних. Частина 2. Структури даних» вивчається на основі знань, отриманих з дисциплін: “Алгоритми та структури даних. Частина 1. Основи алгоритмізації”, “Комп’ютерна дискретна математика”, “Лінійна алгебра та аналітична геометрія”. Вона забезпечує вивчення таких навчальних дисциплін як: “Компоненти програмної інженерії”, “Бази даних”, “Об’єктно-орієнтований аналіз та конструювання програмних систем”.

3. Зміст навчальної дисципліни

Розділ 1. Застосування об'єктно-орієнтованого програмування для обробки структур даних.

Тема 1. Застосування наслідування та поліморфізму.

Тема 2. Застосування інтерфейсів та перелічень для створення структур даних.

Тема 3. Застосування узагальнених типів даних.

Розділ 2. Програмування алгоритмів із застосуванням складних структур даних.

Тема 4. Складні структури даних. Введення в колекції. Динамічні масиви та зв'язні списки.

Тема 5. Структури даних хеш-таблиці та їх реалізація.

Тема 6. Реалізація структур даних черг та стеків.

Тема 7. Реалізація базових структур даних: бінарні дерева.

Тема 8. Множини та відображення. Реалізація колекцій Set та Map.

Тема 9. Структури даних для обробки графів.

Тема 10. Алгоритми обробки орієнтованих та зважених графів.

Тема 11. Алгоритми обчислення найкоротших шляхів у зважених графах.

Розділ 3. Методи розробки алгоритмів із динамічними структурами даних.

Тема 12. Динамічне програмування.

Тема 13. Алгоритми рішення комбінаторних задач.

Тема 14. Потoki вводу-виводу. Робота з файлами.

Тема 15. Методи розробки алгоритмів із поверненням назад, евристичних та переборних алгоритмів.

4. Навчальні матеріали та ресурси

Основна література:

1. Кузьменко І.М., Дацюк О.А. Базові алгоритми та структури даних. Навчальний посібник. К.: КПІ ім. Ігоря Сікорського, 2022. 137 с.
2. Алгоритми та методи обчислень: навчальний посібник / М. А. Новотарський. Київ, КПІ ім. Ігоря Сікорського, 2023. 407 с.
3. Алгоритми та структури даних: конспект лекцій. Частина 2. / Укладачі: О.Д. Воробйова, Л.В. Глазунова. Одеса, ОНАЗ ім. О.С. Попова, 2022. 52 с.

Додаткова література:

1. Крєневич А. Алгоритми і структури даних. Підручник. Київ, ВПЦ "Київський Університет", 2021. 200 с.
2. Герберт Шилдт. Java-8. Повне керівництво. 9-е вид. К.: ООО «В.Д.Віл'ямс», 2022. 1377 с.

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

Розділ 1. Застосування об'єктно-орієнтованого програмування для обробки структур даних	
Тема 1. Застосування наслідування та поліморфізму	
1	Об'єкти як параметри методів. Спадкування, поліморфізм та ключове слово <code>super</code> . Абстрактні класи. Ієрархія успадкування та перетворення типів. Внутрішні класи.
Тема 2. Застосування інтерфейсів та перелічень для створення структур даних	
2	Інтерфейси, основні поняття. Особливості застосування інтерфейсів. Інтерфейси в механізмі зворотного виклику. Перелічення <code>enum</code> , створення структур даних на основі перелічення.
Тема 3. Застосування узагальнених типів даних	
3	Клас <code>Object</code> та його методи. Узагальнені типи й методи: узагальнений клас; конструктор та методи в узагальненому класі; обмеження використання узагальнених типів даних (УДТ); використання декількох УДТ; обмеження узагальнених класів на основі інших класів; обмеження УДТ за допомогою узагальнених класів та інтерфейсів; методи з узагальненими параметрами; узагальнений інтерфейс. Наслідування та узагальнення: базовий узагальнений клас; узагальнений клас-нащадок; перетворення узагальнених типів.
Розділ 2. Програмування алгоритмів із застосуванням складних структур даних	
Тема 4. Складні структури даних. Введення в колекції. Динамічні масиви та зв'язні списки	
4	Принцип дії динамічних масивів, однозв'язних та двозв'язних списків. Огляд різних колекцій та їх реалізація бібліотечними класами. Методи класу <code>ArrayList</code> та інтерфейсу <code>List</code> для обробки динамічних масивів. Методи класу <code>LinkedList</code> для обробки зв'язних списків.
Тема 5. Структури даних хеш-таблиці та їх реалізація	
5	Принцип дії хеш-таблиць. Вирішення колізій. Хеш-функції. Універсальне хешування. Метод відкритої адресації. Подвійне хешування. Класи що реалізують хеш-таблиці. Класи <code>Hashtable</code> та <code>HashMap</code> .
Тема 6. Реалізація структур даних черг та стеків	
6	Порівняння об'єктів за допомогою інтерфейсів <code>Comparable</code> та <code>Comparator</code> . Зберігання об'єктів в колекції черги на основі інтерфейсу <code>Queue</code> . Інтерфейс <code>Deque</code> , що реалізує двонаправлену чергу. Клас <code>ArrayDeque</code> . Застосування класу <code>LinkedList</code> для створення черг та стеків. Клас <code>PriorityQueue</code> .
Тема 7. Реалізація базових структур даних: бінарні дерева	
7	Поняття бінарного дерева. Алгоритми обходу вузлів бінарного дерева: рекурсивний алгоритм обходу в глибину; ітеративний алгоритм обходу дерева в глибину;

	ітеративний алгоритм обходу дерева в ширину. Алгоритми видалення вузлів бінарного дерева.
Тема 8. Множини та відображення. Реалізація колекцій Set та Map	
8	Поняття множини Set. Операції порівняння множин. Алгоритми виконання операцій над множинами. Ітератор колекцій в Java. Інтерфейс Set та класи HashSet, LinkedHashSet, TreeSet. Інтерфейс Map та класи HashMap, LinkedHashMap, TreeMap.
Тема 9. Структури даних для обробки графів	
9	Основні поняття та визначення в теорії графів. Представлення графа в програмі. Перебір вершин графа (обхід графа). Алгоритми обходу вершин графа в глибину та в ширину. Алгоритми побудови мінімального остовного дерева.
Тема 10. Алгоритми обробки орієнтованих та зважених графів	
10	Топологічне сортування графів. Зв'язність в орієнтованих графах. Алгоритм Уоршелла. Зважені графи. Побудова мінімального остовного дерева. Алгоритм Краскала для побудови MST зваженого неорграфа. Алгоритм Прима для побудови MST зваженого неорграфа. Реалізація алгоритмів Краскала та Прима.
Тема 11. Алгоритми обчислення найкоротших шляхів у зважених графах	
11	Обчислення найкоротших шляхів в графі за алгоритмом Дейкстри. Обчислення найкоротших шляхів в графі за алгоритмом Флойда. Порівняльна характеристика алгоритмів обчислення найкоротших шляхів в графі.
Розділ 3. Методи розробки алгоритмів із динамічними структурами даних	
Тема 12. Динамічне програмування	
12	Визначення та поняття динамічного програмування. Застосування динамічного програмування для задачі розрахунку чисел Фібоначчі. Покроковий алгоритм для задач динамічного програмування. Задачі двовимірного динамічного програмування. Огляд прикладних задач, що розв'язуються за допомоги динамічного програмування. Приклади класичних задач, що розв'язуються із застосуванням динамічного програмування: сходження по сходах, пошук розміру найдовшої зростаючої підпослідовності, задача про рюкзак.
Тема 13. Алгоритми рішення комбінаторних задач	
13	Елементи комбінаторики. Перестановки, розміщення, сполучення. Генерація перестановок. Алгоритм Джонсона-Троттера. Алгоритм генерування перестановок за допомоги циклічних зсувів. Алгоритм генерування перестановок у лексикографічному порядку (алгоритм Нарайами). Алгоритм генерації перестановок на основі рекурсії. Алгоритм Хіпа. Алгоритм генерації сполучень (комбінацій). Алгоритм генерації розміщень.
Тема 14. Потoki вводу-виводу. Робота з файлами	
14	Поняття потоків вводу-виводу. Зчитування та запис файлів. FileInputStream та FileOutputStream. Байтові потоки, класи ByteArrayInputStream та ByteArrayOutputStream. Потоки, що буферизуються. Класи BufferedInputStream та BufferedOutputStream. Класи PrintStream та PrintWriter, DataOutputStream та DataInputStream. Зчитування та запис текстових файлів. FileReader та FileWriter.

	Символьні потоки, що буферизуються. <code>BufferedReader</code> та <code>BufferedWriter</code> . Сепіалізація об'єктів. Робота з файлами та папками, клас <code>File</code> . Робота із ZIP-архівами.
Тема 15. Методи розробки алгоритмів із поверненням назад, евристичних та переборних алгоритмів	
15	Метод розробки із поверненням назад. Метод розробки евристичних алгоритмів. Застосування переборних алгоритмів для задач дискретної оптимізації. Застосування різних евристик для зменшення обчислювальних витрат переборних методів.

Лабораторні заняття

№ з/п	Назва теми заняття та перелік основних питань
1	Лабораторне заняття № 1 (Лабораторна робота № 1. Програмування алгоритмів в декількох класах із застосуванням наслідування). Основні питання: створення базового класу, створення полів та методів базового класу, створення класів-спадкоємців, організація доступу до полів та методів, створення об'єктів із урахуванням наслідування.
2	Лабораторне заняття № 2 (Лабораторна робота № 1. Програмування алгоритмів в декількох класах із застосуванням наслідування). Основні питання: використання абстрактних класів та інтерфейсів, перевизначення методів в класах спадкоємців, наслідування властивостей об'єктів від базового класу, виконання завдання на лабораторну роботу, створення Звіту з лабораторної роботи № 1.
3	Лабораторне заняття № 3 (Лабораторна робота № 2. Програмування алгоритмів із застосуванням перелічення). Основні питання: створення перелічень та реалізація перелічень в програмах, можливі приклади перелічень як власного типу даних, побудова перелічення з полями та методами.
4	Лабораторне заняття № 4 (Лабораторна робота № 2. Програмування алгоритмів із застосуванням перелічення). Основні питання: використання оператора вибору <code>switch</code> для обрання значень перелічення, особливості виведення значень констант перелічення, виконання завдання на лабораторну роботу, створення Звіту з лабораторної роботи № 2.
5	Лабораторне заняття № 5 (Лабораторна робота № 3. Програмування алгоритмів із застосуванням однозв'язних та двозв'язних списків). Основні питання: особливості програмування власної структури однозв'язного списку, реалізація методів вставки, видалення, пошуку та виводу елементів однозв'язного списку, особливості програмування двозв'язного списку із можливістю проходження в обох напрямках, порівняння продуктивності базових структур даних: однозв'язних та двозв'язних списків.
6	Лабораторне заняття № 6 (Лабораторна робота № 3. Програмування алгоритмів із застосуванням однозв'язних та двозв'язних списків). Основні питання: використання бібліотечних класів динамічного масиву та двозв'язного списку для програмування списків, особливості використання

	бібліотечних методів роботи зі списками, виконання завдання на лабораторну роботу, створення Звіту з лабораторної роботи № 3.
7	Лабораторне заняття № 7 (Лабораторна робота № 4. Програмування алгоритмів із застосуванням черг та стеку). Основні питання: програмування власного типу даних, що реалізує стек, створення методів: вставки, видалення елемента, та виведення елемента на вершині стеку, програмування власного типу даних, що реалізує чергу, створення методів: вставки, видалення елемента, та виведення елемента на початку черги.
8	Лабораторне заняття № 8 (Лабораторна робота № 4. Програмування алгоритмів із застосуванням черг та стеку). Основні питання: особливості реалізації стеку та черги за допомоги бібліотечних класів, особливості використання бібліотечних методів роботи зі стеком і чергою, особливості організації черги з пріоритетами, виконання завдання на лабораторну роботу, створення Звіту з лабораторної роботи № 4.
9	Лабораторне заняття № 9 (Лабораторна робота № 5. Програмування алгоритмів із застосуванням бінарних дерев). Основні питання: програмування власного типу даних, що реалізує бінарне дерево, створення класу вузла дерева та класу дерева, створення методів: вставки, видалення елемента, створення методів обходів вузлів дерева: в глибину та в ширину, організація бінарного пошуку заданого елемента дерева.
10	Лабораторне заняття № 10 (Лабораторна робота № 5. Програмування алгоритмів із застосуванням бінарних дерев). Основні питання: особливості реалізації бінарного дерева за допомоги бібліотечних класів, особливості використання бібліотечних методів роботи із бінарним деревом, особливості організації балансування бінарного дерева, виконання завдання на лабораторну роботу, створення Звіту з лабораторної роботи № 5.
11	Лабораторне заняття № 11 (Лабораторна робота № 6. Програмування алгоритмів на графах). Основні питання: програмування представлення графа в програмі матрицею або списком суміжності, створення класу вершини графа та класу графа, особливості введення вершин та ребер графа під час його ініціалізації, реалізація методів обходу вершин графа в глибину та в ширину, особливості реалізації методу перевірки зв'язності та знаходження компонент графа, особливості пошуку всіх шляхів графа між заданими вершинами, побудова мінімального остовного дерева за алгоритмами Краскала та Пріма.
12	Лабораторне заняття № 12 (Лабораторна робота № 6. Програмування алгоритмів на графах). Основні питання: особливості використання зважених та орієнтованих графів в програмуванні, реалізація алгоритму Дейкстри для пошуку найкоротшого шляху від заданої вершини до всіх інших, реалізація алгоритму Флойда для пошуку матриці найкоротших шляхів між всіма вершинами, виконання завдання на лабораторну роботу, створення Звіту з лабораторної роботи № 6.

13	Лабораторне заняття № 13 (Лабораторна робота № 7. Програмування алгоритмів динамічного програмування та комбінаторики). Основні питання: особливості програмування задач на основі мемоїзації та табуляції, реалізація алгоритму моделювання руху коника на основі мемоїзації, реалізація алгоритму моделювання руху робота на площині на основі двовимірного динамічного програмування, реалізація алгоритму задачі про рюкзак на основі повного перебору, жадібного алгоритму, динамічного програмування.
14	Лабораторне заняття № 14 (Лабораторна робота № 7. Програмування алгоритмів динамічного програмування та комбінаторики). Основні питання: особливості програмування комбінаторних задач, реалізація алгоритмів генерації перестановок: нерекурсивних та рекурсивних, реалізація алгоритму генерації сполучень, реалізація алгоритму генерації розміщень, виконання завдання на лабораторну роботу, створення Звіту з лабораторної роботи № 7.
15	Лабораторне заняття № 15 (Виконання модульної контрольної роботи). Основні питання: програмування алгоритмів в декількох класах із застосуванням наслідування, із застосуванням перелічення, програмування алгоритмів із застосуванням однозв'язних та двозв'язних списків, черг та стеку, бінарних дерев, програмування алгоритмів на графах, програмування алгоритмів динамічного програмування та комбінаторики.

Практичні заняття

№ з/п	Назва теми заняття та перелік основних питань
1	Практичне заняття № 1. Програмування задач із застосуванням наслідування, абстрактних класів та інтерфейсів. Основні питання: особливості реалізації наслідування від базового класу, створення базового класу, створення полів та методів базового класу, створення класів-спадкоємців, створення об'єктів із урахуванням наслідування, організація доступу до полів та методів, особливості використання абстрактних класів та інтерфейсів, виконання та розбір програм прикладів згідно завдання на практичне заняття № 1.
2	Практичне заняття № 2. Програмування задач із застосуванням узагальнених типів даних. Основні питання: особливості застосування узагальнених типів даних, створення класу або методу із параметризованим типом даних, аналіз переваг використання узагальнених типів даних, особливості обмеження узагальнених типів даних, виконання та розбір програм прикладів згідно завдання на практичне заняття № 2
3	Практичне заняття № 3. Програмування задач із застосуванням однозв'язних та двозв'язних списків. Основні питання: особливості реалізації власних структур даних на основі однозв'язних та двозв'язних списків, реалізація базових операцій додавання та видалення елемента списку, особливості реалізації методу перебору та пошуку заданого елемента в списку, аналіз переваг та недоліків масивів, та списків, виконання та розбір програм прикладів згідно завдання на практичне заняття № 3

4	Практичне заняття № 4. Програмування задач із застосуванням хеш-таблиць та бінарних дерев. Основні питання: особливості програмування задач із застосуванням хеш-таблиць, обробка колізій в хеш-таблицях, особливості застосування бібліотечних класів, що реалізують хеш-таблиці, особливості бібліотечних методів роботи із хеш-таблицями, виконання та розбір програм прикладів згідно завдання на практичне заняття № 4.
5	Практичне заняття № 5. Програмування задач на графах. Основні питання: створення класу вершини графа та класу графа, представлення графа в основному класі програми, особливості ініціалізації графа в програмі із 3 класами, особливості реалізації методів обходу вершин графа в глибину та в ширину, особливості програмування орієнтованих та зважених графів, побудова мінімального остовного дерева за алгоритмами Краскала та Пріма, виконання та розбір програм прикладів згідно завдання на практичне заняття № 5.
6	Практичне заняття № 6. Програмування задач із застосуванням динамічного програмування. Основні питання: розробка алгоритмів розв'язання задач на основі мемоізації та табуляції, розбиття задачі на підзадачі, особливості збереження проміжних результатів обчислення, побудова рекурсивних алгоритмів динамічного програмування, аналіз алгоритмів для прикладних задач, виконання та розбір програм прикладів згідно завдання на практичне заняття № 6.
7	Практичне заняття № 7. Програмування комбінаторних задач. Основні питання: особливості програмування переборних задач, розробка нерекурсивного алгоритму реалізації перестановок, порівняння різних алгоритмів генерації перестановок, особливості застосування алгоритму генерації сполучень, особливості застосування алгоритму генерації розміщень, оцінка алгоритмічної складності комбінаторних алгоритмів, виконання та розбір програм прикладів згідно завдання на практичне заняття № 7.

6. Самостійна робота студента

1	Розділ 1. Застосування об'єктно-орієнтованого програмування для обробки структур даних
	Тема 1. Застосування наслідування та поліморфізму Приклади та реалізація простих списків [3, стор. 16]. Тема 2. Застосування інтерфейсів та перелічень для створення структур даних Приклади та реалізація власних типів даних на основі перелічень [3, стор. 24]. Тема 3. Застосування узагальнених типів даних Приклади та реалізація власних типів даних на узагальнених типів даних [3, стор. 47].

2	Розділ 2. Програмування алгоритмів із застосуванням складних структур даних Тема 4. Складні структури даних. Введення в колекції. Динамічні масиви та зв'язні списки Приклади та реалізація динамічних масивів та списків [1, стор. 12]. Тема 5. Структури даних хеш-таблиці та їх реалізація Приклади та реалізація хеш-таблиць [1, стор. 23]. Тема 6. Реалізація структур даних черг та стеків Приклади та реалізація черг і стеків [1, стор. 35]. Тема 7. Реалізація базових структур даних: бінарні дерева Приклади та реалізація бінарних дерев [1, стор. 48]. Тема 8. Множини та відображення. Реалізація колекцій Set та Map Приклади та реалізація множин та відображень [1, стор. 63]. Тема 9. Структури даних для обробки графів Приклади та реалізація орієнтованих графів [1, стор. 72]. Тема 10. Алгоритми обробки орієнтованих та зважених графів Приклади та реалізація зважених графів [1, стор. 98]. Тема 11. Алгоритми обчислення найкоротших шляхів у зважених графах Приклади та реалізація алгоритмів пошуку мінімальних шляхів [1, стор. 103].
3	Розділ 3. Методи розробки алгоритмів із динамічними структурами даних Тема 12. Динамічне програмування. Реалізація алгоритму динамічного програмування для різних задач [2, стор. 56]. Тема 13. Алгоритми рішення комбінаторних задач. Реалізація алгоритмів комбінаторних задач [2, стор. 68]. Тема 14. Потоки вводу-виводу. Робота із файлами Приклади та реалізація вводу із файлу та виводу в файл [2, стор. 82]. Тема 15. Методи розробки алгоритмів із поверненням назад, евристичних та переборних алгоритмів Реалізація алгоритму з поверненням назад для вирішення завдань [2, стор. 97].

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Для успішного проходження курсу та складання контрольних заходів необхідним є вивчення навчального матеріалу за кожною темою. Специфіка курсу передбачає акцент на розумінні підходів і принципів, отримання практичних навичок, а не просто запам'ятовування визначень. Кожен студент повинен ознайомитися і слідувати Положенню про академічну доброчесність, Статуту і розпорядку дня університету. Для успішного засвоєння програмного матеріалу студент зобов'язаний:

- не запізнюватися на заняття;
- не пропускати заняття, а в разі пропуску відновити за допомогою консультування з викладачем та з використанням Кампус конспект, самостійно вивчити матеріал

пропущеного заняття та скласти відповідні контрольні заходи в індивідуальному порядку;

- конструктивно підтримувати зворотній зв'язок на всіх заняттях;
- брати активну участь у освітньому процесі;
- своєчасно і старанно виконувати завдання для самостійної роботи;
- бути доброзичливим до однокурсників та викладачів;
- брати участь у контрольних заходах;
- за об'єктивних причин (наприклад, хвороба, міжнародне стажування) навчання може відбуватись індивідуально (в дистанційній on-line формі за погодженням із деканом факультету);
- будь-яке копіювання або відтворення результатів чужої праці (у тому числі списування), якщо тільки робота не має груповий формат, використання чужих завантажених з Інтернету матеріалів кваліфікується як порушення норм і правил академічної доброчесності та передбачає притягнення винного до відповідальності, у порядку, визначеному чинним законодавством та Положенням про академічну доброчесність університету. Результатом невиконання та/або недотримання правил може бути оцінка «не зараховано» за курс.

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Максимальна кількість балів з кредитного модуля дорівнює 100.

Рейтинг студента з кредитного модуля складається з балів, отриманих за:

- 1) виконання та захист семи лабораторних робіт;
- 2) виконання завдань на практичних заняттях;
- 3) написання модульної контрольної роботи.

1. Виконання лабораторних робіт

Оцінюються 7 робіт, передбачених робочою програмою. Максимальний ваговий бал – $8 \cdot 7 = 56$ балів

Рейтингові бали кожної роботи складаються з балів за правильність виконання (від 0 до 3) та захист роботи (від 0 до 3), балів за оформлення протоколу роботи (від 0 до 2). За несвоєчасну здачу звіту з лабораторної роботи – штраф 3 бали.

2. Виконання завдань на практичних заняттях

Оцінюється виконання завдань перших 6 практичних занять. Максимальний ваговий бал – $4 \cdot 6 = 24$ бали. Рейтингові бали кожного завдання складаються з балів за правильність виконання (від 0 до 3) та демонстрацію роботи (від 0 до 1). За несвоєчасне виконання завдання – штраф 2 бали.

3. Модульний контроль

Максимальний ваговий бал – 20.

Сума вагових балів контрольних заходів протягом семестру складає:

$$R=56+24+20=100 \text{ балів}$$

Календарний контроль: проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

Необхідною умовою допуску до заліку є зарахування модульної контрольної роботи, а також стартовий рейтинг (r_c) не менше 40% від R , тобто 40 балів.

Студенти, які набрали протягом семестру рейтинг з кредитного модуля менше $0,6R$, зобов'язані виконувати залікову контрольну роботу.

Студенти, які набрали протягом семестру необхідну кількість балів ($RD \geq 0,6R$), мають можливість:

- отримати залікову оцінку (залік) так званим «автоматом» відповідно до набраного рейтингу (таблиця);
- виконувати залікову контрольну роботу з метою підвищення оцінки (у разі отримання оцінки, більшої ніж «автомат» з рейтингу, студент отримує оцінку за результатами залікової роботи). Оцінювання залікової роботи здійснюється за 100-бальною шкалою.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація з дисципліни (освітнього компонента)

Перелік питань, які виносяться на семестровий контроль:

1. Огляд складних структур даних (колекцій).
2. Прості списки. Клас ArrayList та інтерфейс List.
3. 3. Методи класу ArrayList для обробки простих списків.
4. Приклади застосування простих списків.
5. Зв'язані списки. Клас LinkedList.
6. Класифікація та види зв'язних списків.
7. Реалізація однозв'язних лінійних та однозв'язних циклічних списків.
8. Реалізація двозв'язних лінійних та двозв'язних циклічних списків.
9. Методи класу LinkedList для обробки зв'язних списків.
10. Приклади застосування зв'язних списків .
11. Загальні відомості про черги та стек.

12. Реалізація черги та стеку в складних структурах даних.
13. Методи класу ArrayDeque.
14. Застосування черги та стеку для рішення прикладних завдань.
15. Структура даних – хеш-таблиця. Клас HashSet.
16. Структура даних – дерево. Клас TreeSet.
17. Інтерфейси Comparable та Comparator. Використання для сортування.
18. Відображення і клас HashMap.
19. Клас TreeMap.
20. Ітератори та ітерування даних.
21. Потоки вводу-виводу. Закриття потоків.
22. Зчитування та запис файлів. Класи FileInputStream та FileOutputStream.
23. Класи ByteArrayInputStream та ByteArrayOutputStream. Класи BufferedInputStream та BufferedOutputStream.
24. Класи PrintStream та PrintWriter. Класи DataOutputStream та DataInputStream.
25. Зчитування та запис текстових файлів. Класи FileReader, FileWriter, BufferedReader, BufferedWriter.
26. Сериалізація об'єктів.
27. Клас File. Робота з файлами та каталогами. Робота із ZIP-архівами.
28. Клас Console.
29. Загальні відомості та класифікація бінарних дерев.
30. Реалізація бінарних дерев.
31. Основні алгоритми обробки бінарних дерев.
32. Поняття про червоно-чорні дерева та основні алгоритми їх обробки.
33. Основні характеристики графів.
34. Операції над графами.
35. Матричні способи задання графа.
36. Представлення графів як динамічної структури даних.
37. Основні властивості графів. Ізоморфізм графів.
38. Поняття маршрутів в графі.
39. Реалізація в структурах даних маршрутів в графі.
40. Обходи в графах. Ейлерові графи.
41. Зв'язність графа.
42. Зважені графи.
43. Дерева як вид графу. Остовне дерево графа.
44. Пошук мінімального шляху в зваженому орграфі.
45. Пошук мінімального остовного дерева зваженого графа.
46. Поняття про обчислювальну складність алгоритму.
47. Оцінка ресурсів, необхідних на реалізацію алгоритмів.
48. Поняття про NP-повні класи задач. Можливість реалізації алгоритмів для таких задач.
49. Переборні алгоритми та їх реалізація з точки зору обчислювальної складності.
50. Алгоритми розміщення.
51. Алгоритми перестановок.

- 52. Алгоритми сполучення.
- 53. Хешування даних.
- 54. Методи розв'язання колізій.
- 55. Організація даних для прискорення пошуку за вторинними ключами.
- 56. Суть методу динамічного програмування та його особливості.
- 57. Застосування методу динамічного програмування в конкретних задачах
- 58. Метод розробки жадібних алгоритмів.
- 59. Застосовність жадібних алгоритмів. Приклади та їх реалізація.
- 60. Застосування переборних алгоритмів для задач дискретної оптимізації.

**Робочу програму навчальної дисципліни (силабус) «Алгоритми та структури даних.
Частина 2. Структури даних»:**

Складено професором кафедри інженерії програмного забезпечення в енергетиці НН ІАТЕ,
д.т.н., проф. Барабашом Олегом Володимировичем.

Ухвалено кафедрою інженерії програмного забезпечення в енергетиці НН ІАТЕ (протокол
№ 42 від 18.06.2025 р.)

Погоджено Методичною комісією НН ІАТЕ КПІ імені Ігоря Сікорського (протокол №9 від
27.06.2025 р.)