



АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ. ЧАСТИНА 2.

СТРУКТУРИ ДАНИХ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	121 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	4,5 кредити, 135 годин, з яких 72 години аудиторних (36 год лекції, 36 год лабораторні), 63 години 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
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 програм на мові програмування Java 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 1) (*Освітньо-професійна програма першого (бакалаврського) рівня вищої освіти ступеня “бакалавр”. Галузь знань 12 Інформаційні технології. Спеціальність 121 Інженерія програмного забезпечення. Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці*);

- здатність застосовувати знання у практичних ситуаціях (ЗК 2);
- здатність до пошуку, оброблення та аналізу інформації з різних джерел (ЗК 6);
- володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних (ФК 7);
- здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення (ФК 8);
- 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 1);
- 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: “Комп’ютерна дискретна математика”, “Лінійна алгебра та аналітична геометрія”. Вона забезпечує вивчення таких навчальних дисциплін як: “Компоненти програмної інженерії”, “Бази даних”.

Зміст навчальної дисципліни

Розділ 1. Програмування алгоритмів на Java із застосуванням складних структур даних Тема

1. Складні структури даних. Колекції в Java. Прості списки.

Тема 2. Колекції в Java. Зв’язані списки.

Тема 3. Колекції в Java. Черги та стек.

Тема 4. Колекції в Java. Хеш-таблиця та дерева.

Тема 5. Колекції в Java. Відображення.

Тема 6. Серіалізація об'єктів. Робота із файлами та архівами. Тема

7. Алгоритми обробки бінарних дерев.

Тема 8. Структури даних для обробки графів

Тема 9. Алгоритми обчислення структурних характеристик графів

Тема 10. Алгоритми пошуку мінімальних шляхів в зважених графах

Розділ 2. Методи розробки алгоритмів із динамічними структурами даних.

Тема 11. Аналіз алгоритмів. Обчислювальна складність алгоритмів.

Тема 12. Алгоритми рішення комбінаторних задач.

Тема 13. Методи швидкого доступу до даних.

Тема 14. Динамічне програмування.

Тема. 15. Методи розробки жадібних алгоритмів.

Тема. 16. Методи розробки алгоритмів із поверненням назад.

Тема. 17. Методи розробки евристичних та переборних алгоритмів. Тема.

18. Методи розробки імовірнісних алгоритмів.

3. 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, 2019. 407 с.

3. Алгоритми та структури даних: конспект лекцій. Частина 2. / Укладачі: О.Д. Воробйова, Л.В. Глазунова. Одеса, ОНАЗ ім. О.С. Попова, 2017. 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», 2020. 1377 с.

Навчальний контент

4. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

Розділ 1. Програмування алгоритмів на Java із застосуванням складних структур даних	
Тема 1. Складні структури даних. Колекції в Java. Прості списки.	
1 1.	<ul style="list-style-type: none"> 1. Огляд колекцій в Java 2. Прості списки. Клас ArrayList та інтерфейс List 3. Методи класу ArrayList для обробки простих списків. 4. Приклади застосування простих списків.
Тема 2. Колекції в Java. Зв'язані списки.	

2	<ol style="list-style-type: none"> 1. Зв'язані списки. Клас LinkedList 2. Класифікація та види зв'язних списків. 3. Реалізація однозв'язних лінійних та однозв'язних циклічних списків. 4. Реалізація двозв'язних лінійних та двозв'язних циклічних списків. 5. Методи класу LinkedList для обробки зв'язних списків. 6. Приклади застосування зв'язних списків .
Тема 3. Колекції в Java. Черги та стек.	
3	<ol style="list-style-type: none"> 1. Загальні відомості про черги та стек. 2. Реалізація черги та стеку в складних структурах даних. 3. Методи класу ArrayDeque. 4. Застосування черги та стеку для рішення прикладних завдань.
Тема 4. Колекції в Java. Хеш-таблиця та дерева.	
4	<ol style="list-style-type: none"> 1. Структура даних – хеш-таблиця. Клас HashSet 2. Структура даних – дерево. Клас TreeSet 3. Інтерфейси Comparable та Comparator. Використання для сортування.
Тема 5. Колекції в Java. Відображення.	
5	<ol style="list-style-type: none"> 1. Відображення і клас HashMap 2. Клас TreeMap 3. Ітератори
Тема 6. Серіалізація об'єктів. Робота із файлами та архівами.	
6	<ol style="list-style-type: none"> 1. Серіалізація об'єктів 2. Класи ByteArrayInputStream та ByteArrayOutputStream. Класи BufferedInputStream та BufferedOuputStream 3. Класи PrintStream та PrintWriter. Класи DataOutputStream та DataInputStream 4. Клас File. Робота з файлами та каталогами. Робота із ZIP-архівами 5. Клас Console
Тема 7. Алгоритми обробки бінарних дерев.	
7	<ol style="list-style-type: none"> 1. Загальні відомості та класифікація бінарних дерев. 2. Реалізація бінарних дерев. 3. Основні алгоритми обробки бінарних дерев. 4. Основні алгоритми обробки червоно-чорних дерев.
Тема 8. Структури даних для обробки графів	
8	<ol style="list-style-type: none"> 1. Основні характеристики графів. 2. Операції над графами 3. Матричні способи задання графа. 4. Представлення графів як динамічної структури даних. 5. Основні властивості графів. Ізоморфізм графів. 6. Маршрути в графі.
Тема 9. Алгоритми обчислення структурних характеристик графів	

9	<ol style="list-style-type: none"> 1. Маршрути в графі. 2. Обходи в графах. Ейлерові графи. 3. Зв'язність графа.
Тема 10. Алгоритми пошуку мінімальних шляхів в зважених графах	
10	<ol style="list-style-type: none"> 1. Зважені графи 2. Дерева. 3. Пошук мінімального шляху в зваженому орграфі 4. Пошук мінімального остовного дерева зваженого графа.
Розділ 2. Методи розробки алгоритмів із динамічними структурами даних.	
Тема 11. Аналіз алгоритмів. Обчислювальна складність алгоритмів.	
11	<ol style="list-style-type: none"> 1. Поняття про обчислювальну складність алгоритму. 2. Оцінка ресурсів, необхідних на реалізацію алгоритмів. 3. Поняття про NP-повні класи задач. Можливість реалізації алгоритмів для таких задач. 4. Переборні алгоритми та їх реалізація з точки зору обчислювальної складності.
Тема 12. Алгоритми рішення комбінаторних задач.	
12	<ol style="list-style-type: none"> 1. Алгоритми розміщення. 2. Алгоритми перестановок. 3. Алгоритми сполучення.
Тема 13. Методи швидкого доступу до даних.	
13	<ol style="list-style-type: none"> 1. Хешування даних. 2. Методи розв'язання колізій. 3. Організація даних для прискорення пошуку за вторинними ключами.
Тема 14. Динамічне програмування.	
14	<ol style="list-style-type: none"> 1. Суть методу динамічного програмування та його особливості 2. Застосування методу в конкретних задачах
Тема. 15. Методи розробки жадібних алгоритмів.	
15	<ol style="list-style-type: none"> 1. Метод розробки жадібних алгоритмів. 2. Застосовність жадібних алгоритмів. 3. Приклади та їх реалізація.
Тема. 16. Методи розробки алгоритмів із поверненням назад.	
16	<ol style="list-style-type: none"> 1. Метод розробки алгоритмів із поверненням назад. 2. Приклади та їх реалізація.
Тема. 17. Методи розробки евристичних та переборних алгоритмів.	
17	<ol style="list-style-type: none"> 1. Метод розробки евристичних алгоритмів. 2. Застосування переборних алгоритмів для задач дискретної оптимізації. 3. Застосування різних евристик для зменшення обчислювальних витрат переборних методів. 4. Приклади та їх реалізація.

Тема. 18. Методи розробки імовірнісних алгоритмів.	
18	1. Метод програмування випадкових величин за різними законами розподілу. 2. Метод розробки імовірнісних алгоритмів. 3. Приклади та їх реалізація.

Лабораторні заняття

1	Програмування алгоритмів із застосуванням однозв'язних та двозв'язних списків
2	Програмування алгоритмів із застосуванням черг та стеку.
3	Програмування алгоритмів із застосуванням бінарних дерев
4	Програмування алгоритмів задач на графах
5	Програмування задач обробки комбінаторних об'єктів
6	Програмування задач із застосуванням динамічного програмування
7.	Програмування задач на розв'язання графів. Динамічне програмування. Робота з файлами

5. Самостійна робота студента

1	Розділ 1. Програмування алгоритмів на Java із застосуванням складних структур даних
	Тема 1.1. Складні структури даних. Колекції в Java. Прості списки. <i>Приклади та реалізація простих списків [4, стор. 312].</i> Тема 1.2. Колекції в Java. Зв'язані списки. <i>Приклади та реалізація зв'язаних списків [4, стор. 348].</i> Тема 1.3. Колекції в Java. Черги та стек. <i>Приклади та реалізація черг і стеків [4, стор. 352].</i> Тема 1.4. Колекції в Java. Хеш-таблиця та дерева. <i>Приклади та реалізація хеш-таблиць та дерев [4, стор. 358].</i> Тема 1.5. Колекції в Java. Відображення <i>Приклади та реалізація відображень [4, стор. 368].</i>
	Тема 1.6. Потоки вводу-виводу. Робота із файлами <i>Приклади та реалізація вводу із файлу та виводу в файл [4, стор. 382].</i> Тема 1.7. Алгоритми обробки бінарних дерев. <i>Приклади та реалізація бінарних дерев [4, стор. 404].</i> Тема 1.8. Структури даних для обробки графів <i>Приклади та реалізація орієнтованих графів [4, стор. 428].</i> Тема 1.9. Алгоритми обчислення структурних характеристик графів. <i>Приклади та реалізація зважених графів [4, стор. 448].</i> Тема 1.10. Алгоритми пошуку мінімальних шляхів в зважених графах <i>Приклади та реалізація алгоритмів пошуку мінімальних шляхів [4, стор. 463].</i>
2	Розділ 2. Методи розробки алгоритмів із динамічними структурами даних

<p>Тема 2.1. Аналіз алгоритмів. Обчислювальна складність алгоритмів. Методи обчислення складності алгоритмів [4, стор. 508].</p> <p>Тема 2.2. Алгоритми рішення комбінаторних задач. Реалізація алгоритмів комбінаторних задач [4, стор. 534].</p> <p>Тема 2.3. Методи швидкого доступу до даних. Порівняльний аналіз методів швидкого доступу до даних [4, стор. 548].</p> <p>Тема 2.4. Динамічне програмування. Реалізація алгоритму динамічного програмування [4, стор. 566].</p> <p>Тема 2.5. Методи розробки жадібних алгоритмів. Реалізація жадібних алгоритмів на прикладах [4, стор. 616].</p> <p>Тема 2.6. Методи розробки алгоритмів із поверненням назад. Реалізація алгоритму з поверненням назад [4, стор. 637].</p> <p>Тема 2.7. Методи розробки евристичних та переборних алгоритмів. Реалізація евристичних та переборних алгоритмів для вирішення практичних завдань [4, стор. 682].</p> <p>Тема 2.8. Методи розробки імовірнісних алгоритмів. Реалізація імовірнісних алгоритмів для вирішення практичних завдань [4, стор. 682].</p>

Політика та контроль

6. 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.

7. 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) роботу на лекційних заняттях;
- 2) виконання та захист семи лабораторних робіт;
- 3) написання модульної контрольної роботи.

1.Робота на лекційних заняттях

На лекційних заняттях може бути проведено бліцопитування студентів щодо виконання домашнього завдання. Такі опитування проводяться на довільних лекційних заняттях 10 разів протягом семестру, на початку заняття. Ваговий бал за вірну відповідь – 1. Максимальна кількість балів, що може отримати кожен студент за семестр – 10 балів.

2.Виконання лабораторних робіт

Оцінюються 7 робіт, передбачених робочою програмою. Максимальний ваговий бал – $10 \cdot 7 = 70$ балів

Рейтингові бали кожної роботи складаються з балів за правильність виконання (від 0 до 4) та захист роботи (від 0 до 4), балів за оформлення протоколу роботи (від 0 до 2). За несвоєчасну здачу звіту з лабораторної роботи – штраф 5 балів.

3. Модульний контроль

Максимальний ваговий бал – 20.

Сума вагових балів контрольних заходів протягом семестру складає:

$$R = 10 + 70 + 20 = 100 \text{ балів}$$

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

Необхідною умовою допуску до заліку є зарахування модульної контрольної роботи, а також стартовий рейтинг (rc) 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:

<i>Кількість балів</i>	<i>Оцінка</i>
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

8. Додаткова інформація з дисципліни (освітнього компонента)

Перелік питань, які виносяться на семестровий контроль:

1. Огляд колекцій в Java
2. Прості списки. Клас ArrayList та інтерфейс List 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 оргграфі.
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. Метод розробки алгоритмів із поверненням назад. Приклади та їх реалізація.
61. Метод розробки евристичних алгоритмів.
62. Застосування переборних алгоритмів для задач дискретної оптимізації.
63. Застосування різних евристик для зменшення обчислювальних витрат переборних методів.
64. Методи програмування випадкових величин за різними законами розподілу.
65. Методи розробки імовірнісних алгоритмів.

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

Складено професором кафедри ІПЗЕ, д.т.н., проф. Барабашом Олегом Володимировичем

Ухвалено кафедрою ІПЗЕ (протокол № 28 від 15.05.2023 р.)

Погоджено Методичною комісією ННІАТЕ КПІ ім. Ігоря Сікорського ¹ (протокол № 9 від 26.05.2023 р.)

¹ [Методичною радою університету – для загальноуніверситетських дисциплін.](#)