|  |  |  |
| --- | --- | --- |
|  |  | **Кафедра обчислювальної техніки** |
| **Алгоритми та структури даних 2.**  **Структури даних**  **Робоча програма навчальної дисципліни (Силабус)** | | |

# Реквізити навчальної дисципліни

|  |  |
| --- | --- |
| Рівень вищої освіти | *Перший (бакалавр)* |
| Галузь знань | *12 Інформаційні технології* |
| Спеціальність | *121 Інженерія програмного забезпечення* |
| Освітня програма | *Інженерія програмного забезпечення* |
| Статус дисципліни | *Нормативна* |
| Форма навчання | *очна* |
| Рік підготовки, семестр | *1 рік ( 2 семестр)* |
| Обсяг дисципліни | *4,5 кредитів (ECTS). Виділення часу -135 годин, включаючи 72 аудиторних години та 63 годин самостійного навчання.* |
| Семестровий контроль/ контрольні заходи | *2 семестр – залік* |
| Розклад занять | *2 заняття на тиждень, розклад на* [*http://rozklad.kpi.ua/*](http://rozklad.kpi.ua/) |
| Мова викладання | *Українська* |
| Інформація про  керівника курсу / викладачів | Лектор та викладач з практичних занять магістр науковий: *Анастасія Анатоліївна Молчанова, email an.ser.313kpi@gmail.com* |
| Розміщення курсу | Лекційний матеріал: *https://bbb.comsys.kpi.ua/b/ana-yt9-2hw*  Методичні рекомендації: *https://kanyevsky.kpi.ua/студентам/алгоритми-і-структури-даних/* |

# Програма навчальної дисципліни

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

Викладання алгоритмів та структур даних в університеті має такі аспекти:

Академічний аспект. Алгоритми та структури даних - це наука, яка дає фундаментальні знання. Мета - ознайомити студентів з основними поняттями та методами представлення даних та виконання з ними алгоритмів, вимірюючи складність алгоритму.

Професійний аспект. Знання структур даних та алгоритмів є обов’язковими для інженерів та програмістів, які розробляють програмне забезпечення для комп’ютера. Мета цього курсу – сфор­мувати навички використання мови С для програмування базових алгоритмів, які використову­ються у більшості програмного забезпечення універсальних та спеціалізованих систем.

Інтелектуальні та освітні аспекти. Вивчення структур даних та алгоритмів сприяє розвитку когнітивних навичок, що є суттєвим інтелектуальним фактором у процесі створення програмного забезпечення.

В результаті вивчення цієї дисципліни досягаються такі результати навчання.

Компетенції:

• здатність до абстрактного мислення, аналізу та синтезу;

• здатність до навчання та набуття сучасних знань;

• вміння застосовувати знання в практичних ситуаціях;

• здатність до алгоритмічного та логічного мислення

• вміння оформляти отримані результати роботи у вигляді презентацій, науково -технічних звітів.

• здатність ідентифікувати, класифікувати та формулювати вимоги до

програмного забезпечення

•володіння знаннями про інформаційні моделі даних, здатність створювати

програмне забезпечення для зберігання, видобування та опрацювання даних

Знання:

• основні поняття та методи теорії алгоритмів;

• основні структури даних та алгоритми;

• базові знання про архітектуру комп’ютера, операційну систему, компіляцію програм, обчислювальні процеси в комп’ютерах;

• знати підходи щодо оцінки та забезпечення якості програмного забезпечення

• методи вимірювання складності алгоритму та програми.

Навички:

• програмування базових алгоритмів мовою C з різними структурами даних;

• вирішення задач аналізу складності алгоритмів та програм;

• знати та вміти використовувати методи та засоби збору, формулювання та

аналізу вимог до програмного забезпечення

• систематично мислити та застосовувати творчі здібності для формування нових ідей.

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

Передумови, тобто дисципліни, вивчення яких має передувати вивченню цієї дисципліни:

• базовий рівень володіння англійською мовою не нижче A1.

Постреквізити - дисципліни, вивченню яких має передувати вивчення даної дисципліни:

• архітектура комп’ютера;

• бази даних;

• розробка програм;

• системне програмне забезпечення;

• чисельні методи;

• комп’ютерні мережі;

• штучний інтелект.

**3. Зміст навчальної дисципліни**

Тема 1: Рекурсивні функції.

Рекурсивна підпрограма та її реалізація. Рекурсивна функція в математичних обчисленнях. Переваги та недоліки рекурсії.

Тема 2: Пов’язані списки.

Динамічне виділення пам'яті. Функції для роботи з динамічною пам'яттю. Купа. Приклад програмування циклічного буфера. Забезпечення послідовності розподілу динамічної пам'яті. Ініціалізація динамічних змінних.

Послідовності. Мультимножина. Вектор Дерево. Граф. Операції з множинами. Бітова шкала. Послідовне зберігання векторів. Зв’язане зберігання векторів. Лінійний список та його програмування. Зв'язані структури даних. Пошук у зв’язаному списку. Операції зі списком. Різновиди зв’язаних списків.

Стек і черга. Реалізація стека в архітектурі рівня системи команд. Стек як зв'язаний список. Сортування даних у стеках. Мови програмування, орієнтовані на стек. Циклічний буфер як черга. Список як черга. Черга з пріоритетом. Дек.

Тема 3: Алгоритми на графах.

Визначення графів. Диграфи. Типи графів та їх параметри. Маршрути у графах. Граф Ейлера. Гамільтоновий цикл. Частини графа. Подільність діграфа. Дерева і ліс. Кістяк. Плоскі графи.

Графічні структури даних. Графічне прикладне програмування.

Відношення та графи. Перехідне та рефлексивне замикання. Матриця досяжності. Компонент сильного зв'язку. Граф конденсації. Алгоритм Кьоніга.

Обхід графу. Алгоритм пошуку вшир. Алгоритм обчислення відстаней. Алгоритм пошуку вглиб. Топологічне сортування. Пошук вершини з’єднання.

Зважений граф. Найкоротший шлях. Алгоритм Дейкстри. Жадібні алгоритми оптимізації. Мінімальний пошук по кістяку. Алгоритм Краскала. Алгоритм Пріма. Пошук дерева Штейнера.

Орієнтовані дерева. Упорядковані дерева. Визначення виразів, програм за впорядкованими деревами. Прямий обхід бінарних дерев. Представлення дерев у програмах. Представлення вільних дерев. Алгоритми коду Прюфера. Представлення двійкового дерева. Структура списку, упакований масив, польська нотація. Алгоритми обходу двійкових дерев. Польські прямаі та зворотна нотації. Дерево сортування.

Тема 4: Асоціативна пам’ять

Основи асоціативної пам’яті. Операції з асоціативною пам'яттю. Асоціативні масиви та списки та їх програмування. Алгоритм операцій з деревом сортування. Вирівняні та повні дерева. Збалансовані дерева. ABL-дерево. Алгоритми балансування дерев.

Хеш -таблиця та її структура. Створення асоціативної хеш -таблиці. Пошук об’єкта у таблиці за допомогою хеш -функції. Тип даних абстрактної хеш -таблиці. Модульне хешування. Ідеальна хеш -функція. Методи вирішення конфліктів.

**4. Навчальні матеріали**

Базова література

1. Б. Керниган, Д. Рітчі. Мова програмування Сі. 229 с.

2. Кормен Т. Г., Лейзерсон Ч. Е., Рівест Р. Л., Стайн К. Вступ до алгоритмів. Київ: К. І. С., 2019. 1288 с.

3. Дж. Макконелл. Основи сучасних алгоритмів. 2004. 368 с.

Додаткова література

1. Н. Н. Непийвода, І. Н. Скопін. Основи програмування. 2002. 919 с.

2. Н. Вірт, Алгоритми та структури даних. 358 с. ДМК. 2010. 272 с.

Слайди лекцій та керівництва до лабораторних робіт <https://kanyevsky.kpi.ua/en/for-students/>

**Обладнання, що необхідне для проведення занять**

Лекційні заняття проводяться в аудиторії, яку обладнано проектором, практичні заняття – в комп’ютерному класі.

# Навчальний контент

**5. Методика опанування навчальної дисципліни (освітнього компонента)**

Навчальний зміст дисципліни складається з лекцій та лабораторних робіт.

**Лекції.**

Тема 1: Рекурсивні функції.

Лекція 1. Рекурсивні алгоритми.

Математична індукція та рекурсія. Рекурсивна підпрограма та її реалізація. Рекурсивна функція. Сигналізуюча функція. Шлях рекурсії. 3 форми рекурсивних дій. Функція з рекурсивним підйомом або спуском. Функція, з частковим спуском і підйомом. Обчислення чисел Фібоначчі. Рекурсивна функція в математичних обчисленнях. Переваги та недоліки рекурсії.

Тема 2: Зв’язані списки.

Лекція 2. Динамічна адресація.

Розподіл статичної пам’яті. Динамічне виділення пам'яті. Тип size\_t. Функції для роботи з динамічною пам'яттю. Купа. Приклад програмування циклічного буфера. Забезпечення послідовності розподілу динамічної пам'яті. Ініціалізація динамічних змінних.

Лекція 3. Зв’язані структури даних.

Послідовності. Множини. Вектор Дерево. Граф. Гіперграф. Операції з множинами. Бітова шкала. Послідовне зберігання векторів. Зв’язане зберігання векторів. Зв’язаний список. Вузол списку. Лінійний список та його програмування.

Лекція 4. Зв’язані структури даних (продовження)

Пошук у зв’язаному списку. Видалення елементу зі списку. Додавання елементу до відсортованого списку. З’єднання списків. Переваги та недоліки подання списку. Різновиди зв’язаних списків. Порожній список. Список із заголовком. Циклічний список. Двохзв'язний список. Дввохзв'язний циклічний список. Список з пропусками. Список з самоорганізацією.

Лекція 5. Стек і черга.

Визначення стека. Операції зі стеком. Реалізація стека в архітектурі системи команд. Карта пам'яті звичайного застосунку. Перевірка переповнення стека. Недоліки організації стека в масиві. Стек як зв'язаний список. Використання стека. Сортування даних у стеках. Стек у граматичному аналізі текстів. Мови програмування, орієнтовані на стек. Стекові процесори. Визначення черги. Циклічний буфер як черга. Список як черга. Черга з пріоритетом. Дек.

Тема 3: Алгоритми на графах.

Лекція 6. Графи.

Визначення графу. Призначення графів. Напрямлений граф. Параметри графів. Види графів. Зважені графи. Ізомрофні графи. Маршрути у графах. Граф Ейлера. Гамільтоновий цикл. Орієнтовані маршрути. Частини графа. З'єднання з графіком. Подільність діграфа. Дерева і ліс. Кістяк графа. Плоскі графи. Графи Понтрягіна - Куратовського. Гомеоморфні графи.

Лекція 7. Структури графічних даних.

Графічний інтерфейс комп’ютера і його основні елементи. Графічний інтерфейс у PC-AT. Багатозадачна операційна система. Інтерфейс графічного пристрою (GDI). Бібліотека windows.h. Дескриптор HANDLE. Графічне прикладне програмування. Основний модуль WinMain (). Приклад програмування фреймового застосунку. Деталі виконання третьої лабораторної роботи.

Лекція 8. Графи та відношення.

Рефлексивні, антирефлексивні, симетричні, антисиметричні, зворотні, універсальні, взаємодоповнюючі відношення. Об'єднання графів. Композиція відношень. Степінь відношення. Транзитивне відношення. Транзитивне та рефлексивне замикання. Матриця досяж­ності. Компоненти сильної зв'язності. Матриця сильної зв'язності. Граф конденсації. Алгоритм Кеніга.

Лекція 9. Алгоритми обходу графів.

Обхід графів. Алгоритм пошуку вшир. Процедура BFS. Протокол обходу графа. Властивості BFS. BFS-дерево та розрахунок відстані. Геодезичне дерево. Алгоритм обчислення відстаней.

Лекція 10. Алгоритми обходу графів (продовження).

Алгоритм пошуку вглиб. Процедура DFS. Роль стека. Приклад реалізації алгоритму DFS. Властивості алгоритму DFS. DFS-дерево. Топологічна сортування. Спільний точковий пошук.

Лекція 11. Алгоритми найкоротшого шляху.

Зважений граф. Найкоротший шлях. Алгоритм Дейкстри. Жадібні алгоритми оптимізації. Приклад виконання алгоритму Дейкстри.

Лекція 12. Пошук мінімального кістяка.

Мінімальний кістяк. Задачі з мінімальним кістяком. Алгоритм Краскала. Приклад виконання алгоритму Краскала. Алгоритм Пріма. Приклад виконання алгоритму Пріма. Порівняння алгоритмів. Використання алгоритмів мінімального кістяка. Пошук дерева Штейнера.

Лекція 13. Дерева.

Визначення орієнтованого дерева. Частини дерев. Впорядковані дерева. Визначення виразів, програм за впорядкованими деревами. Техніка відступів. Довідники як впорядковані дерева. Обхід дерева. Двійкові дерева. Представлення дерев у програмах. Представлення вільних дерев. Код Прюфера. Алгоритми кодування та декодування коду Прюфера.

Лекція 14. Дерева (продовження).

Двійкові дерева. Представлення двійкового дерева. Представлення двійкового дерева в програмах: структура списку, упакований масив, польська нотація. Алгоритми обходу двійкових дерев: прямі, внутрішні, обернені. Польські прямі та зворотні нотації. Дерево сортування.

Тема 4: Асоціативна пам’ять

Лекція 15. Основи асоціативної пам’яті.

Асоціативна пам’ять. Операції з асоціативною пам'яттю. Асоціативні типи пам’яті. Асоціативні масиви і списки та їх програмування. Порівняння представлень асоціативної пам’яті.

Лекція 16. Пошук у двійковому дереві

Алгоритм пошуку в дереві сортування. Алгоритм вставки в дерево сортування. Алгоритм видалення з дерева сортування. Порівняння представлень асоціативної пам’яті. Вирівняні та повні дерева. Збалансовані дерева. ABL-дерево. Алгоритми балансування дерев.

Лекція 17. Хеш -таблиці

Створення асоціативної таблиці. Пошук в таблиці за допомогою хеш-функції. Хеш-таблиця та її структура. Тип даних абстрактної хеш-таблиці. Хешування ключів. Модульне хешування. Хешу­вання ключа з плаваючою крапкою. Хешування рядків. Ідеальна хеш-функція. Властивості ефектив­ної хеш-функції. Методи вирішення конфліктів. Хешування з відкритою адресацією. Хешування методом виділення та створення ланцюжка. Порівняння представлень асоціативної пам’яті.

Лекція 18.

Модульна контрольна робота.

**Лабораторні роботи**

Лабораторна робота 1.

Рекурсивні алгоритми.

Лабораторна робота 2.

Програмування зв'язаного списку.

Лабораторна робота 3.

Графічне представлення графів.

Лабораторна робота 4.

Характеристики графа та його відношень.

Лабораторна робота 5.

Алгоритми обходу графів.

Лабораторна робота 6.

Алгоритми мінімального кістяка.

**6. Самостійна робота студента**

Самонавчання включає самостійну роботу студентів і полягає в наступному:

• підготовка до лекцій шляхом вивчення попереднього лекційного матеріалу, а також літературних джерел, на яких ґрунтується матеріал попередніх лекцій (перелік джерел та перелік розділів подаються разом із матеріалом лекції);

• підготовка до лабораторної вправи шляхом ознайомлення із завданням та методичними вказівками до лабораторних робіт, включаючи вивчення теоретичного матеріалу, необхідного для відповіді на контрольні запитання до лабораторних робіт;

Кожна лабораторна робота вимагає для підготовки та виконання двох тижнів.

# Політика та контроль

**7. Політика навчальної дисципліни (освітнього компонента)**

Система вимог до студентів:

• студент зобов’язаний відвідувати лекції та лабораторні заняття, активно працювати над засвоєнням викладеного на них матеріалу;

• на лекції лектор використовує власний презентаційний матеріал;

• лабораторні роботи захищаються у два етапи: перший етап - студенти виконують завдання, складають електронний звіт та надсилають викладачу; другий етап - захист лабораторних робіт у лабораторії. Контроль знань на лабораторних роботах здійснюється шляхом перевірки звіту про лабораторні роботи, а також питань з матеріалів робіт.

• модульна контрольна робота пишеться як тест на лекції з використанням усіх наявних матеріалів, але без використання комп’ютерних засобів.

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

На першому занятті студенти знайомляться з політикою оцінювання, яка ґрунтується на Положенні про систему оцінювання результатів навчання https://document.kpi.ua/files/2020\_1-273.pdf. Рейтинг студента на курсі складається з балів, які він/вона отримує за захищені лабораторні роботи (R1) та модульну контрольну роботу (R2).

Rs = R1+R2 = 100 балів

В результаті максимальний середній бал дорівнює:

6 лабораторних робіт x 10 балів = 60 балів

модульна контрольна робота = 40 балів.

Рейтинг вважається дійсним за умови, що студент здав усі 6 лабораторних робіт.

Згідно з університетськими правилами щодо моніторингу успішності студента (https://kpi.ua/document\_control), існує два тижні оцінювання (атестація), зазвичай протягом 7/8 та 14/15 тижнів семестру, коли оцінюється прогрес у навчанні студентів та відповідність до критеріїв політики оцінювання курсу.

Умовою першої атестації є отримання не менше 10 балів (на момент атестації). Умова другої атестації - набрати не менше 30 балів (на момент атестації).

Критеріями оцінки є:

• Виконання лабораторних робіт:

- бездоганна робота оцінюється у 9 балів;

- є певні недоліки в оформленій роботі - 8-7 балів;

- є деякі недоліки у виконанні робочої програми - 6-5 балів;

Робота не виконана або не захищена - 0 балів.

За роботу, подану вчасно, діє заохочення - 1 бал (сумарно не більше - 6 балів).

• Модульна контрольна робота оцінюється 40 балами. Контрольна робота складається з 16 тестових питань, а також практичного завдання (скласти програму) зі списку, наведеного в додатку до робочої програми.

За кожну правильну відповідь на тестове питання нараховується 2 бали. Відповідь на практичне завдання оцінюється 8 балами за такими критеріями:

- "відмінно" - правильний текст програми з коментарями - 8 - 7 балів;

- "добре" - текст програми, загалом правильний, але не використаний прийом оптимізації - 6-5 балів;

- "задовільно" - є деякі принципові помилки в тексті програми - 4 - 3 бали;

- "незадовільно" - незадовільна відповідь - 0 балів.

Студенти, які нарешті наберуть необхідну кількість балів (≥60), можуть:

- отримати остаточну залікову оцінку відповідно до рейтингового балу;

- виконати залікову контрольну роботу, щоб підвищити оцінку.

Студенти можуть отримати до 6 заохочувальних балів за виконання творчих робіт з кредитного модуля (складання тез, участь у конкурсах, у дослідженнях тощо).

Студенти, чий остаточний бал за успішність Rs нижче 60 балів, але більше за 30, повинні виконати залікову контрольну роботу. Студенти, чий остаточний бал за успішність Rs вище або дорівнює 60 балів, можуть виконати залікову контрольну роботу, щоб підвищити оцінку. В обох випадках застосовується вимога - попередній рейтинг студента складає: 0.7R1. Студенти, чий бал нижче 30, не допускаються до заліку.

Максимальна оцінка R3 за залікову контрольну роботу складає 60 балів. Контрольна робота складається з 25 тестових питань, а також практичного завдання (скласти програму). За кожну правильну відповідь на тестове питання нараховується 2 бали. Відповідь на практичне завдання оцінюється 10 балами за такими критеріями:

- "відмінно" - правильний текст програми з коментарями - 10 - 9 балів;

- "добре" - текст програми, загалом, правильний, але не використані прийоми оптимізації - 8-6 балів;

- "задовільно" - є деякі принципові помилки в тексті програми - 5 - 4 балів;

- "незадовільно" - незадовільна відповідь - 0 балів.

Рейтингова оцінка разом з оцінкою за залікову контрольну роботу дорівнює

Rs = 0.7R1 + R3.

Таблиця. Остаточна оцінка RS приймається за системою оцінювання університету

|  |  |
| --- | --- |
| *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 |

**Робочу програму навчальної дисципліни (силабус):**

**Складено** д.т.н., с.н.с, А. М. Сергієнко

**Ухвалено кафедрою обчислювальної техніки** (протокол №13 від 10.05.2023)

**Погоджено Методичною комісією ФІОТ** (протокол №11 від 29.06.2023)