



АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ. Частина 1.

ОСНОВИ АЛГОРИТМІЗАЦІЇ

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	Обов'язкова (нормативна)
Форма навчання	Очна (денна)
Рік підготовки, семестр	1 курс, осінній семестр
Обсяг дисципліни	4 кредити, 120 годин, з яких 68 години аудиторних (30 год. лекції, 30 год. лабораторні, 8 год. практичні), 52 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: Макарчук Андрій Валентинович, andreymakarh2@gmail.com 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 1. Основи алгоритмізації» 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 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, робота 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 даних. Частина 1. Основи алгоритмізації» 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 1. Основи алгоритмізації» не має дисциплін, які її забезпечують, вона вивчається на основі знань, отриманих у середній загальноосвітній школі. 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. Програмування розгалужених та циклічних алгоритмів.

Тема 1. Вступ до алгоритмів та структур даних.

Тема 2. Прості структури даних. Змінні та операції.

Тема 3. Програмування розгалужених алгоритмів.

Тема 4. Масиви та їх обробка із застосуванням циклічних алгоритмів.

Розділ 2. Програмування алгоритмів обробки інформації.

Тема 5. Алгоритми матричних перетворень.

Тема 6. Алгоритми пошуку елементів масивів. Рекурсія.

Тема 7. Алгоритми сортування масивів.

Тема 8. Програмування класів, об'єктів, методів.

Тема 9. Програмування із застосуванням конструкторів класів.

Тема 10. Алгоритми обробки текстової інформації.

Тема 11. Відлагодження та тестування програм.

Тема 12. Обробка виключних ситуацій.

Розділ 3. Структуризація алгоритмів в класах та методах.

Тема 13. Структуризація алгоритмів. Поняття інкапсуляції.

Тема 14. Перевантаження методів та поняття поліморфізму.

Тема 15. Якість та ефективність програм.

4. Навчальні матеріали та ресурси

Основна література:

1. Кублій Л.І. Алгоритми та структури даних. Основи алгоритмізації. Підручник. К.: КПІ ім. Ігоря Сікорського, 2022. 528 с.

<https://ela.kpi.ua/handle/123456789/48282>

2. Алгоритми та методи обчислень: навчальний посібник / М. А. Новотарський. Київ, КПІ ім. Ігоря Сікорського, 2023. 407 с.

3. Стратієнко Н.К., Годлевський М.Д., Бородіна І.О. Алгоритми і структури даних: практикум. Навчальний посібник. Харків, НТУ «ХП», 2025. 224 с.

Додаткова література:

4. Крєневич А. Алгоритми і структури даних. Підручник. Київ, ВПЦ "Київський Університет", 2025. 200 с.

5. Герберт Шилдт. Java-8. Повне керівництво. 9-е вид. К.: ООО «В.Д.Віл'ямс», 2023. 1377 с.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Лекційні заняття

Розділ 1. Програмування розгалужених та циклічних алгоритмів	
Тема 1. Вступ до алгоритмів та структур даних.	
1	Історія розвитку програмування. Місце дисципліни в системі підготовки фахівців. Вимоги до студентів. Рейтингова система оцінювання. Особливості алгоритмізації на різних мовах програмування. Найпростіша програма.
Тема 2. Прості структури даних. Змінні та операції.	
2	Типи даних та змінні в мовах програмування. Примітивні типи. Особливості застосування літералів. Оголошення та ініціалізація змінних. Перетворення і приведення типів даних. Операції (арифметичні, побітові, операції відношення, логічні, присвоєння, умовна операція, пріоритети операцій). Особливості вводу-виводу.
Тема 3. Програмування розгалужених алгоритмів.	
3	Програмування розгалужених алгоритмів. Розгалуження з оператором if-else. Розгалуження з оператором switch. Особливості використання операторів вибору для розгалуження алгоритмів. Створення власних функцій. Бібліотеки класів.
Тема 4. Масиви та їх обробка із застосуванням циклічних алгоритмів.	
4	Масиви. Ітераційні оператори (циклу). Оператори переходу break, continue, return. Приклади використання циклів. Правила побудови блок-схем алгоритмів.
Розділ 2. Програмування алгоритмів обробки інформації	
Тема 5. Алгоритми матричних перетворень.	
5	Особливості доступу до елементів матриць та їх перебору. Алгоритми додавання та множення матриць. Алгоритми рішення систем лінійних алгебраїчних рівнянь за методом Гауса.
Тема 6. Алгоритми пошуку елементів масивів. Рекурсія.	
6	Алгоритми пошуку елементів масивів. Загальна характеристика методів пошуку. Алгоритм лінійного пошуку. Алгоритм бінарного пошуку. Особливості обробки масивів за допомогою класу Arrays. Особливості рекурсивного виклику методів.
Тема 7. Алгоритми сортування масивів.	
7	Базові алгоритми сортування масивів. Сортування вибором мінімального елемента. Сортування методом бульбашки. Сортування методом вставки. Сортування вибором. Сортування злиттям. Швидке сортування. Порівняльний аналіз методів сортування.
Тема 8. Програмування класів, об'єктів, методів.	
8	Основні відомості про класи. Порядок створення об'єктів. Методи. Додавання метода в клас. Повернення із метода. Повернення значення. Використання параметрів. Додавання параметризованого метода в клас.

Тема 9. Програмування із застосуванням конструкторів класів.	
9	Конструктори. Конструктори без параметрів. Параметризовані конструктори. Додавання конструктора в клас. Особливості оператора new. Ключове слово this.
Тема 10. Алгоритми обробки текстової інформації.	
10	Використання рядків (текстових змінних) для обробки текстової інформації. Створення рядків. Методи роботи з рядками. Масиви рядків. Властивість незмінності рядків. Використання рядків для управління оператором switch. Особливості стандартних потоків вводу та виводу.
Тема 11. Відлагодження та тестування програм.	
11	Відлагодження програм. Найпростіші прийоми відлагодження. Використання протоколювання для відлагодження. Інструментальні засоби відлагодження. Тестування програм. Роль тестування в програмному проєкті. Модульне тестування за допомоги JUnit.
Тема 12. Обробка виключних ситуацій.	
12	Введення в обробку виключень. Оператор throws. Класи виключень. Створення своїх класів виключень. Робота з текстовими файлами: зчитування та запис.
Розділ 3. Структуризація алгоритмів в класах та методах.	
Тема 13. Структуризація алгоритмів. Поняття інкапсуляції.	
13	Структуризація програмних модулів в пакетах. Управління доступом до полів та методів класу. Модифікатори доступу та інкапсуляція. Поняття інкапсуляції.
Тема 14. Перевантаження методів та поняття поліморфізму.	
14	Передача об'єктів методам. Способи передачі аргументів методу. Повернення об'єктів із методів. Перевантаження методів та конструкторів. Застосування ключового слова static. Статичні блоки. Вкладені та внутрішні класи. Використання методів із змінним числом аргументів. Перевантаження таких методів. Поняття поліморфізму.
Тема 15. Якість та ефективність програм.	
18	Поняття про якість програмного забезпечення. Основні показники ефективності. Дослідження часу виконання фрагментів коду. Приклад: порівняння двох способів отримання рядка з примітивного типу. Приклад: порівняння інтерпретації та синхронної компіляції. Поняття про Java Code Conventions.

Лабораторні заняття

№ з/п	Назва теми заняття та перелік основних питань
1	Лабораторне заняття № 1 (Лабораторна робота № 1. Програмування лінійних алгоритмів). Основні питання: створення простих програм обчислення формул з дробовими та цілими даними, введення програми в текстовий файл, компілювання програми, створення виконавчого файлу, виконання програми, усунення помилок в тексті програми.

2	Лабораторне заняття № 2 (Лабораторна робота № 1. Програмування лінійних алгоритмів). Основні питання: налаштування середовища програмування IntelliJ Idea, ознайомлення з елементами панелі управління, створення проекту, пакету, класу, виконання найпростішої програми, використання форматowanego виведення даних в консоль.
3	Лабораторне заняття № 3 (Лабораторна робота № 1. Програмування лінійних алгоритмів). Основні питання: використання в програмі різних типів даних, приведення та перетворення типів даних; виконання завдання на лабораторну роботу, створення Звіту з лабораторної роботи № 1.
4	Лабораторне заняття № 4 (Лабораторна робота № 2. Програмування розгалужених алгоритмів). Основні питання: створення алгоритмів із розгалуженнями на основі умовних операторів if, If-else та оператора вибору switch, використання логічних операцій NOT, AND, OR в розгалужених алгоритмах.
5	Лабораторне заняття № 5 (Лабораторна робота № 2. Програмування розгалужених алгоритмів). Основні питання: розробка алгоритмів на основі вкладених та складних умовних конструкцій, виконання завдання на лабораторну роботу, створення Звіту з лабораторної роботи № 2.
6	Лабораторне заняття № 6 (Лабораторна робота № 3. Програмування циклічних алгоритмів). Основні питання: розробка алгоритмів з використанням операторів циклів із заданим числом ітерацій, з передумовою, з постумовою, використання операторів break, continue в циклічних алгоритмах.
7	Лабораторне заняття № 7 (Лабораторна робота № 3. Програмування циклічних алгоритмів). Основні питання: розробка алгоритмів з використанням вкладених циклів, виконання завдання на лабораторну роботу, створення Звіту з лабораторної роботи № 3.
8	Лабораторне заняття № 8 (Лабораторна робота № 4. Програмування задач пошуку та сортування масивів даних). Основні питання: ініціалізація масивів, особливості вводу та виводу елементів масивів, реалізація алгоритмів лінійного пошуку, реалізація алгоритмів бінарного пошуку, реалізація алгоритму пошуку мінімального та максимального елементів масиву.
9	Лабораторне заняття № 9 (Лабораторна робота № 4. Програмування задач пошуку та сортування масивів даних). Основні питання: реалізація алгоритмів сортування елементів масиву різними способами: бульбашки, вибором мінімального елементу, вставки, виконання завдання на лабораторну роботу, створення Звіту з лабораторної роботи № 4.
10	Лабораторне заняття № 10 (Лабораторна робота № 5. Програмування алгоритмів обробки текстових рядків). Основні питання: розробка алгоритмів з використанням методів класу String для обробки текстів, переведення текстової змінної в масив символів і навпаки, розбиття текстової змінної на слова, пошук в тексті заданого символу, слова.

11	Лабораторне заняття № 11 (Лабораторна робота № 5. Програмування алгоритмів обробки текстових рядків). Основні питання: розробка алгоритмів виділення текстового підрядка, порівняння текстових рядків, особливості введення текстових рядків з консолі, статистична обробка текстів, виконання завдання на лабораторну роботу, створення Звіту з лабораторної роботи № 5.
12	Лабораторне заняття № 12 (Лабораторна робота № 6. Програмування обробки матриць та розв'язання систем лінійних алгебраїчних рівнянь). Основні питання: розробка алгоритмів виконання операцій над двома матрицями, реалізація окремого класу для матриці з методами додавання, множення та транспонування, обчислення детермінанта матриці.
13	Лабораторне заняття № 13 (Лабораторна робота № 6. Програмування обробки матриць та розв'язання систем лінійних алгебраїчних рівнянь). Основні питання: розробка алгоритму розв'язання систем лінійних алгебраїчних рівнянь методом Крамера, розробка алгоритму розв'язання систем лінійних алгебраїчних рівнянь методом Гауса, виконання завдання на лабораторну роботу, створення Звіту з лабораторної роботи № 6.
14	Лабораторне заняття № 14 (Лабораторна робота № 7. Моделювання взаємодії об'єктів із застосуванням двох класів). Основні питання: розробка алгоритмів із застосуванням класу-шаблону опису об'єктів, особливості використання в класі полів, конструкторів та методів, особливості створення об'єктів.
15	Лабораторне заняття № 15 (Лабораторна робота № 7. Моделювання взаємодії об'єктів із застосуванням двох класів). Основні питання: розробка алгоритмів моделювання декількох об'єктів, виконання порівняння об'єктів за певними полями, використання властивості інкапсуляції для закриття полів, використання гетерів і сетерів для доступу до приватних полів, виконання завдання на лабораторну роботу, створення Звіту з лабораторної роботи № 7.

Практичні заняття

№ з/п	Назва теми заняття та перелік основних питань
1	Практичне заняття № 1. Розробка простих програм з лінійними алгоритмами. Основні питання: розробка програм з лінійними алгоритмами з обробки даних різних типів, правила запису арифметичних виразів, виконання операцій над даними різних типів, введення даних з клавіатури, виконання та розбір програм прикладів згідно завдання на практичне заняття № 1.
2	Практичне заняття № 2. Розробка програм на основі розгалужених та циклічних алгоритмів. Основні питання: особливості використання умовних операторів if, If-else, використання оператора вибору switch, використання логічних операцій NOT, AND, OR, особливості використання операторів циклів із заданим числом ітерацій, з передумовою, з

	постумовою, використання операторів break, continue під час циклічних обчислень, особливості використання вкладених циклів, виконання та розбір програм прикладів згідно завдання на практичне заняття № 2.
3	Практичне заняття № 3. Розробка програм із застосуванням класів, об'єктів, методів. Основні питання: розробка алгоритмів із застосуванням додаткового класу, що описує об'єкти, особливості використання в додатковому класі полів, конструкторів та методів, особливості створення об'єктів, особливості доступу до полів та методів, порівняння об'єктів за заданим полем, виконання та розбір програм прикладів згідно завдання на практичне заняття № 3.
4	Практичне заняття № 4. Розробка програм обробки текстової інформації та створення модульних тестів. Основні питання: використання методів класу String для обробки текстів, переведення текстової змінної в масив символів і навпаки, розбиття текстової змінної на слова, пошук в тексті заданого символу, слова, порівняння текстових рядків, відлагодження програми інструментальними засобами IntelliJ Idea, створення JUnit-тестів, виконання та розбір програм прикладів згідно завдання на практичне заняття № 4.

6. Самостійна робота студента

1	Розділ 1. Програмування розгалужених та циклічних алгоритмів
	Тема 1. Вступ до алгоритмів та структур даних. Компіляція Java-програми [1, стор. 41]. Тема 2. Прості структури даних. Змінні та операції. Область дії та час існування змінних [1, стор. 77]. Тема 3. Програмування розгалужених алгоритмів. Особливості застосування оператора break [1, стор. 84]. Тема 1.4. Масиви та їх обробка із застосуванням циклічних алгоритмів. Перетворення та приведення типів даних [1, стор. 88].
2	Розділ 2. Програмування алгоритмів обробки інформації
	Тема 5. Алгоритми матричних перетворень. Методи обробки n-вимірних масивів [1, стор. 93]. Тема 6. Алгоритми пошуку елементів масивів. Рекурсія. Алгоритм бінарного пошуку [1, стор. 98]. Тема 7. Алгоритми сортування масивів. Порівняльний аналіз методів сортування [1, стор. 106]. Тема 8. Програмування класів, об'єктів, методів. Збирання сміття та методи завершення [1, стор. 167]. Тема 9. Програмування із застосуванням конструкторів класів. Ключове слово this [1, стор. 171]. Тема 10. Алгоритми обробки текстової інформації. Особливості стандартних потоків вводу та виводу [1, стор. 121].

	Тема 11. Відлагодження та тестування програм. Модульне тестування за допомоги JUnit [1, стор. 132]. Тема 12. Обробка виключних ситуацій. Обробка виключень під час роботи з файлами [1, стор. 148].
3	Розділ 3. Структуризація алгоритмів в класах та методах
	Тема 13. Структуризація алгоритмів. Поняття інкапсуляції. Використання методів: сетерів і гетерів для приватних членів класу [1, стор. 204]. Тема 14. Перевантаження методів та поняття поліморфізму. Використання методів із змінним числом аргументів. [1, стор. 328]. Тема 15. Якість та ефективність програм. Поняття про Java Code Conventions [1, стор. 363].

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 крайніх 3 практичних занять. Максимальний ваговий бал – $3 \cdot 8 = 24$ бали. Рейтингові бали кожного завдання складаються з балів за правильність виконання (від 0 до 5) та демонстрацію роботи (від 0 до 3). 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:

<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

9. Додаткова інформація з дисципліни (освітнього компонента)

Перелік питань, які виносяться на семестровий контроль:

1. Історія розвитку програмування.
2. Відмінності Java від C++
3. Примітивні типи даних
4. Особливості застосування літералів
5. Оголошення та ініціалізація змінних
6. Звичайна та динамічна ініціалізація змінних.
7. Область видимості та час існування змінних.
8. Перетворення і приведення типів даних
9. Явне та автоматичне перетворення типів даних
10. Програмування розгалужених алгоритмів із застосуванням оператора if-else. Вкладені конструкції if-else.
11. Програмування розгалужених алгоритмів із застосуванням оператора switch
12. Особливості застосування операторів break, default при застосуванні оператора switch.
13. Особливості використання операторів вибору для розгалуження алгоритмів
14. Види операцій. Умовна операція.
15. Арифметичні та побітові операції.
16. Операції відношення, логічні операції.
17. Операції присвоєння, умовна операція.
18. Пріоритети операцій.
19. Масиви. Методи оголошення та ініціалізації одновимірних масивів.
20. Масиви. Методи оголошення та ініціалізації багатовимірних масивів.
21. Доступ до елементів масиву. Методи перебору елементів масивів
22. Ітераційні оператори (цикли).
23. Особливості застосування циклів із передумовою та післяумовою.
24. Особливості застосування операторів переходу break, continue, return.
25. Приклади використання циклів.
26. Основні відомості про класи та об'єкти. Визначення класу.
27. Порядок створення об'єктів. Доступ до полів об'єкту.
28. Особливості присвоєння та порівняння змінних посилального (об'єктного) типу.
29. Правила створення методів в класі.
30. Модифікатори доступу.
31. Використання параметрів, що передаються в метод.
32. Особливості повернення результатів роботи методів.
33. Особливості використання параметризованих та непараметризованих методів.
34. Застосування конструкторів в класах.
35. Особливості використання параметризованих та непараметризованих конструкторів.

36. Особливості оператора new.
37. Особливості застосування ключового слова this.
38. Особливості типу String.
39. Способи створення текстових рядків.
40. Методи роботи з рядками
41. Особливості створення масивів рядків.
42. Особливості присвоєння та порівняння рядків.
43. Основні методи класу String.
44. Особливості застосування масиву символів та текстових рядків.
45. Методи перебору символів текстового рядку.
46. Властивість незмінності рядків
47. Використання рядків для управління оператором switch
48. Особливості рекурсивного виклику методів
50. Найпростіші прийоми відлагодження.
51. Використання протоколювання для відлагодження.
52. Інструментальні засоби відлагодження.
53. Тестування програм та його роль в програмному проекті.
54. Модульне тестування за допомоги JUnit.
55. Особливості стандартних потоків вводу та виводу.
56. Особливості рекурсивного виклику методів
57. Побудова програми з декількома класами.
58. Використання об'єктів, як екземплярів класів.
59. Використання гетерів і сетерів.
60. Структуризація алгоритмів в класах, об'єктах і методах.

Робочу програму навчальної дисципліни (силабус) «Алгоритми та структури даних. Частина 1. Основи алгоритмізації»:

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 № 43 від 24.06.2026 р.)

Погоджено Методичною комісією НН ІАТЕ КПІ імені Ігоря Сікорського (протокол № 9 від 26.06.2026 р.)