



# АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ. ЧАТИНА 1. ОСНОВИ АЛГОРИТМІЗАЦІЇ

## 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	Заочна
Рік підготовки, семестр	1 курс, осінній семестр
Обсяг дисципліни	4 кредити, 120 годин, з яких 8 годин аудиторних (4 год лекції, 4 год лабораторні), 112 годин становить самостійна робота
Семестровий контроль/ контрольні заходи	Залік/модульна контрольна робота.
Розклад занять	<a href="http://rozklad.kpi.ua/">http://rozklad.kpi.ua/</a>
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: д.т.н., проф. Барабаш Олег Володимирович, <a href="mailto:bar64@ukr.net">bar64@ukr.net</a> Лабораторні заняття: д.т.н., проф. Барабаш Олег Володимирович, <a href="mailto:bar64@ukr.net">bar64@ukr.net</a>
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 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 01) (*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 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 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. Основи програмування на мові Java

Тема 1. Вступ до програмування на Java

Тема 2. Прості структури даних. Операції в Java.

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. Класи, об'єкти, методи в мові Java.

Тема 9. Конструктори класів в мові Java

Тема 10. Алгоритми обробки текстової інформації

Тема 11. Відлагодження та тестування програм

Тема 12. Обробка виключень в мові Java

#### Розділ 3. Структуризація алгоритмів в класах та методах.

Тема 13. Управління доступом до даних. Перевантаження методів.

Тема 14. Наслідування та поліморфізм в Java.

Тема 15. Інтерфейси в Java. Перелічення.

Тема 16. Клас Object, узагальнення та клонування об'єктів

Тема 17. Застосування класів та інтерфейсів для вирішення практичних завдань.

Тема 18. 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, 2019. 407 с.

3. Стратієнко Н.К., Годлевський М.Д., Бородіна І.О. Алгоритми і структури даних: практикум. Навчальний посібник. Харків, НТУ «ХП», 2021. 224 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

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

#### Лекційні заняття

<b>Розділ 1. Основи мови програмування на мові Java</b>	
<b>Тема 1. Вступ до програмування на Java.</b>	
1	Історія створення мови Java. Особливості мови Java. Відмінності Java від C++. Найпростіша програма на Java. Hello World!.
<b>Тема 2. Прості структури даних. Операції в Java.</b>	
2	Типи даних та змінні в Java. Примітивні типи. Особливості застосування літералів в Java. Оголошення та ініціалізація змінних. Перетворення і приведення типів даних. Операції в Java (арифметичні, побітові, операції відношення, логічні, присвоєння, умовна операція, пріоритети операцій). Особливості вводу-виводу.

**Примітка.** Теоретичний матеріал за лекціями № 3 – 18 опановується під час міжзборового періоду

#### Лабораторні заняття

1	Програмування лінійних алгоритмів
2	Програмування розгалужених алгоритмів

**Примітка.** Лабораторні роботи № 3 – 7 виконуються під час міжзборового періоду

### 6. Самостійна робота студента

1	<p><b>Розділ 1. Основи програмування на мові Java</b></p> <p><b>Тема 1.1.</b> Вступ до програмування на Java. Компіляція Java-програми [1, стор. 41].</p> <p><b>Тема 1.2.</b> Прості структури даних. Програмування розгалужених алгоритмів. Область дії та час існування змінних [1, стор. 77].</p> <p><b>Тема 1.3.</b> Програмування розгалужених алгоритмів. Програмування розгалужених алгоритмів. Розгалуження з оператором if-else. Розгалуження з оператором switch. Особливості використання операторів вибору для розгалуження алгоритмів. Створення власних функцій. Бібліотеки класів. Особливості застосування оператора break [1, стор. 84].</p> <p><b>Тема 1.4.</b> Масиви та їх обробка із застосуванням циклічних алгоритмів. Масиви. Ітераційні оператори (циклу). Оператори переходу break, continue, return. Приклади використання циклів. Правила побудови блок-схем алгоритмів. Перетворення та приведення типів даних [1, стор. 88].</p>
2	<b>Розділ 2. Програмування алгоритмів обробки інформації</b>

	<p><b>Тема 2.1.</b> Алгоритми матричних перетворень. Особливості доступу до елементів матриць та їх перебору. Алгоритми додавання та множення матриць. Алгоритми рішення систем лінійних алгебраїчних рівнянь за методом Гауса. Методи обробки n-вимірних масивів [1, стор. 93].</p> <p><b>Тема 2.2.</b> Алгоритми пошуку елементів масивів. Рекурсія. Алгоритми пошуку елементів масивів. Особливості рекурсивного виклику методів. Алгоритм бінарного пошуку. Алгоритм бінарного пошуку [1, стор. 98].</p> <p><b>Тема 2.3.</b> Алгоритми сортування масивів. Базові алгоритми сортування масивів. Сортування вибором мінімального елемента. Сортування методом бульбашки. Сортування методом вставки. Сортування вибором. Сортування злиттям. Швидке сортування. Порівняльний аналіз методів сортування. Порівняльний аналіз методів сортування [1, стор. 106].</p> <p><b>Тема 2.4.</b> Класи, об'єкти, методи в мові Java. 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. Збирання сміття та методи завершення [1, стор. 167].</p> <p><b>Тема 2.5.</b> Конструктори класів в мові Java. Конструктори. Конструктори без параметрів. Параметризовані конструктори. Додавання конструктора в клас. Особливості оператора new. Ключове слово this. [1, стор. 171].</p> <p><b>Тема 2.6.</b> Алгоритми обробки текстової інформації. Використання рядків (строк) в Java 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. [1, стор. 121].</p> <p><b>Тема 2.7.</b> Відлагодження та тестування програм. Відлагодження програм. Найпростіші прийоми відлагодження. Використання протоколювання для відлагодження. Інструментальні засоби відлагодження. Тестування програм. Роль тестування в програмному проекті. Модульне тестування за допомоги JUnit [1, стор. 132].</p>
	<p><b>Тема 2.8.</b> Обробка виключень в мові Java. Введення в обробку виключень. Оператор throws. Класи виключень. Створення своїх класів виключень. [1, стор. 158].</p>
3	<p><b>Розділ 3. Структуризація алгоритмів в класах та методах</b></p>
	<p><b>Тема 3.1.</b> Управління доступом до даних. Перевантаження методів. Управління доступом до полів та методів класу. Модифікатори доступу в Java. 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. [1, стор. 204].</p> <p><b>Тема 3.2.</b> Наслідування та поліморфізм в Java.</p>

	<p>Об'єкти як параметри методів. Спадкування, поліморфізм та ключове слово <code>super</code>. Абстрактні класи. Ієрархія успадкування та перетворення типів. Внутрішні класи [1, стор. 328].</p> <p><b>Тема 3.3.</b> Інтерфейси в Java. Перелічення.</p> <p>Інтерфейси. Інтерфейси в механізмі зворотного виклику. Перелічення <code>enum</code> [1, стор. 223].</p> <p><b>Тема 3.4.</b> Потоки вводу-виводу. Робота із файлами.</p> <p>Потоки вводу-виводу. Закриття потоків. Зчитування та запис файлів. Класи <code>FileInputStream</code> та <code>FileOutputStream</code>. Зчитування та запис текстових файлів з обробкою виключень. Класи <code>FileReader</code>, <code>FileWriter</code>, <code>BufferedReader</code>, <code>BufferedWriter</code>. [1, стор. 265].</p> <p><b>Тема 3.5.</b> Застосування класів та інтерфейсів для вирішення практичних завдань.</p> <p>Приклади застосування наслідування класів. Приклади застосування інтерфейсів.</p> <p>Ієрархія класів. Розробка програм із декількома класами. [1, стор. 326].</p> <p><b>Тема 3.6.</b> Якість та ефективність програм.</p> <p>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 <code>Java Code Conventions</code> [1, стор. 332].</p>
4	<b>Виконання Лабораторної роботи № 3</b> Програмування циклічних алгоритмів.
5	<b>Виконання Лабораторної роботи № 4</b> Програмування алгоритмів обробки текстових рядків.
6	<b>Виконання Лабораторної роботи № 5</b> Програмування задач пошуку та сортування масивів даних.
7	<b>Виконання Лабораторної роботи № 6</b> Програмування обробки матриць із застосуванням декількох класів.
8	<b>Виконання Лабораторної роботи № 7</b> Програмування розв'язання систем лінійних алгебраїчних рівнянь

## 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) написання модульної контрольної роботи.

### 1. 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.

### 2. Модульний контроль

Максимальний ваговий бал – 20.

Сума вагових балів контрольних заходів протягом семестру складає:

$$R = 70 + 30 = 100 \text{ балів}$$

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 створення мови Java. Особливості мови Java.
2. Відмінності Java від C++
3. Примітивні типи даних в Java
4. Особливості застосування літералів в Java
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 в Java. Умовна операція в Java.
15. Арифметичні та побітові операції в Java.
16. Операції відношення, логічні операції в Java
17. Операції присвоєння, умовна операція
18. Пріоритети операцій в Java
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) в Java.
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 стандартних потоків вводу та виводу.
49. 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

**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 №34 від 10.05.2024 р.)

**Погоджено** Методичною комісією НН ІАТЕ КПІ імені Ігоря Сікорського (протокол №9 від 31.05.2024 р.)