

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ  
УНІВЕРСИТЕТ ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

**ЗАТВЕРДЖЕНО**

на засіданні кафедри  
інформатики та комп'ютерної техніки  
Протокол № 21 від 29.08.2025 р.

**ПОГОДЖЕНО**

Проректор з навчально-методичної  
роботи



Каріна ШЕМАШКАЛО

**ОСНОВИ АЛГОРИТМІЗАЦІЇ**

**робоча програма навчальної дисципліни (РПНД)**

Галузь знань	<b>Ф "Інформаційні технології"</b>
Спеціальність	<b>Ф6 "Інформаційні системи і технології"</b>
Освітній рівень	<b>перший (бакалаврський)</b>
Освітня програма	<b>"Штучний інтелект"</b>
Статус дисципліни	<b>обов'язкова</b>
Мова викладання, навчання та оцінювання	<b>українська</b>

Розробник:

к.т.н., доц.

Олексій  
ГОРОХОВАТСЬКИЙ

Завідувач кафедри  
інформатики та комп'ютерної  
техніки

Сергій УДОВЕНКО

Гарант програми

Сергій УДОВЕНКО

**Харків  
2025**

## ВСТУП

Вивчення та знання основ алгоритмізації відносяться до одних з фундаментальних компетентностей для майбутнього розробника комп'ютерних програм. Розуміння теорії створення алгоритмів дозволяє глибше розуміти, як працюють комп'ютерні програми, оптимізувати їх виконання та створювати ефективні рішення складних задач. Знання та розуміння базових алгоритмів допомагає розробнику обрати найбільш підходящий алгоритм для конкретної задачі, що призводить до кращої продуктивності та ефективності коду, знаходити вузькі місця в коді та способи їх усунення, розробляти складні програмні системи (наприклад, системи штучного інтелекту). Загалом, вивчення основ алгоритмізації розвиває логічне мислення та навички розв'язання проблем, що дозволяє майбутнім програмістам ефективно справлятися з новими завданнями. Таким чином, формування компетентностей щодо використання основних алгоритмів та розуміння принципів їх роботи є важливим.

Навчальна дисципліна "Основи алгоритмізації" є обов'язковою навчальною дисципліною, яка вивчається відповідно до навчального плану підготовки здобувачів за спеціальністю Ф6 "Інформаційні системи та технології" першого (бакалаврського) рівня усіх форм навчання. Програму навчальної дисципліни розроблено у відповідності до вимог галузевого стандарту вищої освіти на базі освітньо-професійної програми підготовки бакалавра.

Метою навчальної дисципліни є формування у майбутніх фахівців з інформаційних систем та технологій системи компетентностей з питань розроблення, аналізу та імплементації алгоритмів та аналізу їх якості.

Завданнями навчальної дисципліни є:

- засвоєння основних понять та підходів до розроблення та реалізації алгоритмів;
- вивчення методів аналізу ефективності алгоритмів.

Предметом навчальної дисципліни є відомі алгоритми та алгоритмічні підходи для обробки даних.

Об'єктом навчальної дисципліни є процес розроблення, реалізації та аналізу алгоритмів.

Навчальна дисципліна "Основи алгоритмізації" знайомить здобувачів з основними визначеннями теорії алгоритмів, властивостями алгоритмів, способами їх представлення. Увагу приділено основам програмування мовою C#, що дозволяє реалізувати та аналізувати алгоритми на практиці. В матеріалах курсу розглянуто популярні класичні алгоритми пошуку, обчислення агрегатних значень, роботи із масивами, матрицями, алгоритми сортування. Розглядаються відомі підходи до розробки алгоритмів та проблема визначення складності алгоритмів.

Програма навчальної дисципліни передбачає навчання у формі лекцій, лабораторних занять та самостійної роботи здобувачів. Для практичного засвоєння основних тем дисципліни лабораторні заняття, індивідуальна робота та консультації проводяться з застосуванням персональних комп'ютерів, локальної

мережі та мережі Інтернет у комп'ютерних класах. Всі види занять забезпечуються необхідними електронними методичними матеріалами.

З метою підвищення ефективності вивчення навчальної дисципліни здобувачі мають змогу користуватись системою дистанційного навчання ХНЕУ ім. С. Кузнеця.

Результати навчання та компетентності, які формує навчальна дисципліна, визначено в табл. 1.

Таблиця 1

Результати навчання та компетентності, які формує навчальна дисципліна

Результати навчання	Компетентності, якими повинен оволодіти здобувач вищої освіти
ПР 2	ІК, КЗ 1, КЗ 2, КЗ 5, КЗ 6, КЗ 8, КЗ 10, КС 4
ПР 3	ІК, КЗ 8, КС 4

де:

ІК. Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми в області інформаційних систем та технологій, або в процесі навчання, що характеризуються комплексністю та невизначеністю умов, які потребують застосування теорій та методів інформаційних технологій.

ПР 2. Застосовувати знання фундаментальних і природничих наук, системного аналізу та технологій моделювання, стандартних алгоритмів та дискретного аналізу при розв'язанні задач проектування і використання інформаційних систем та технологій.

ПР 3. Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання інформаційних систем та технологій.

КЗ 1. Здатність до абстрактного мислення, аналізу та синтезу.

КЗ 2. Здатність застосовувати знання у практичних ситуаціях.

КЗ 5. Здатність вчитися і оволодівати сучасними знаннями.

КЗ 6. Здатність до пошуку, оброблення та узагальнення інформації з різних джерел.

КЗ 8. Здатність оцінювати та забезпечувати якість виконуваних робіт.

КЗ 10. Здатність зберігати та примножувати моральні, культурні, наукові цінності і досягнення суспільства на основі розуміння історії та закономірностей розвитку предметної області, її місця у загальній системі знань про природу і суспільство та у розвитку суспільства, техніки і технологій, використовувати різні види та форми рухової активності для активного відпочинку та ведення здорового способу життя

КС 4. Здатність проектувати, розробляти та використовувати засоби реалізації інформаційних систем, технологій та інфокомунікацій (методичні, інформаційні, алгоритмічні, технічні, програмні та інші).

# ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

## Зміст навчальної дисципліни

### Змістовий модуль 1. Основи алгоритмізації.

#### Тема 1. Вступ в алгоритмізацію.

1.1. Мета, завдання, актуальність, важливість дисципліни. Структура дисципліни та навчальних матеріалів. Особливості процесу навчання та вимоги до виконання завдань.

1.2. Алгоритмізація vs програмування.

1.3. Інформація та дані. Історія розвитку комп'ютерних та інформаційних технологій.

1.4. Визначення алгоритму. Властивості алгоритмів. Алан Тьюринг.

1.5. Представлення алгоритмів. Текстовий, аналітичний, графічний опис, псевдокод.

1.6. Базові типи алгоритмів.

#### Тема 2. Введення в мову програмування C#.

2.1. Обґрунтування вибору мови програмування. Загальні відомості про .NET та C#. Середовище розробки (IDE). Онлайн-компілятори. Програма та виконання програми.

2.2. "Hello, world" console application, його розробка та розбір. Спеціальні символи та правила найменування. Синтаксис мови. Змінні та типи даних. Введення та виведення даних. Перетворення типів. Оператори мови програмування. Читання даних з файлу.

2.3. Умовний оператор та особливості використання.

2.4. Масиви та цикли. Багатовимірні масиви.

2.5. Функції та передача параметрів.

2.6. Структури даних: стек, черга, список, словник.

#### Тема 3. Деякі популярні алгоритми.

3.1. Перестановка значень змінних.

3.2. Лінійний пошук.

3.3. Обчислення суми та середнього значення.

3.4. Пошук мінімального та максимального значення.

3.5. Бінарний пошук.

3.6. Пошук мінімального та максимального значень у матриці.

3.7. Сума рядків та стовпців.

### Змістовий модуль 2. Алгоритмічні підходи та оцінка алгоритмів.

#### Тема 4. Алгоритмічні підходи до розробки.

4.1. Випадкові числа та швидкодія.

4.2. Порівняння чисел.

4.3. Повний перебір: недоліки та переваги, приклади використання.

4.4. Рекурсивні алгоритми.

4.5. "Розділяй і володарюй" (divide and conquer).

4.6. Жадібні алгоритми.

4.7. Динамічне програмування.

4.8. Підхід "гілок та меж" (branch and bound).

4.9. Евристичні алгоритми.

**Тема 5. Складність алгоритмів.**

5.1. Аналіз швидкості виконання алгоритму. O-, o-,  $\theta$ -нотації.

5.2. Задачі класів P та NP.

5.3. Приклади складності розглянутих алгоритмів.

5.4. Задача комівояжера.

**Тема 6. Алгоритми сортування.**

6.1. Властивості алгоритмів сортування, принципи порівнянь та вибору відповідних алгоритмів.

6.2. Сортування «бульбашкою». Сортування вставками, шейкерне сортування.

6.3. Швидке сортування. Сортування злиттям.

Перелік лабораторних занять за навчальною дисципліною наведено в табл. 2

Таблиця 2

### Перелік лабораторних занять

Назва теми та завдання	Зміст
Тема 1-2. Лабораторна робота 1. Завдання 1	Розробка та імплементація алгоритмів, що мають лінійну структуру
Тема 2-3. Лабораторна робота 2. Завдання 2	Розробка та імплементація алгоритмів, що мають умовний оператор
Тема 2-3. Лабораторна робота 3. Завдання 3	Розробка та імплементація алгоритмів з циклами
Тема 4-6. Лабораторна робота 4. Завдання 4	Розробка та імплементація циклічних алгоритмів для роботи з масивами
Тема 4-6. Лабораторна робота 5. Завдання 5	Розробка та імплементація циклічних алгоритмів для роботи з матрицями

Перелік самостійної роботи за навчальною дисципліною наведено в табл. 3.

Таблиця 3

### Перелік самостійної роботи

Назва теми	Зміст
Тема 1	Дослідження та розробка найпростіших алгоритмів, які можуть виконувати обчислення, введення та виведення даних
Тема 2-3	Ознайомлення із відомими алгоритмами та розуміння їх реалізацій та принципів роботи
Тема 4	Дослідження та розробка найпростіших алгоритмів, які можуть виконувати обчислення, аналіз із використанням умовних операторів обробляти масиви даних циклами
Тема 5	Дослідження оцінювання алгоритмів з точки зору обчислювальної складності
Тема 6	Дослідження та порівняння різних методів сортування даних

Кількість годин лекційних, лабораторних занять та годин самостійної роботи наведено в робочому плані (технологічній карті) з навчальної дисципліни.

## МЕТОДИ НАВЧАННЯ

У процесі викладання навчальної дисципліни для набуття визначених результатів навчання, активізації освітнього процесу передбачено застосування таких методів навчання, як:

- словесні (теми 1-6), елементи проблемних лекції (теми 1-6);
- наочні (демонстрації (теми 1-6));
- практичні (лабораторні заняття (теми 1-6)).

В умовах змішаної форми навчання подання лекційного матеріалу та/або проведення лабораторних занять та групових та індивідуальних консультацій відбувається з використанням платформи Zoom, в умовах звичайної аудиторної форми заняття проводяться очно, в аудиторіях та комп'ютерних залах.

## ФОРМИ ТА МЕТОДИ ОЦІНЮВАННЯ

Університет використовує 100 бальну накопичувальну систему оцінювання результатів навчання здобувачів вищої освіти.

**Поточний контроль** здійснюється під час проведення лекційних та лабораторних занять і має на меті перевірку рівня підготовленості здобувача вищої освіти до виконання конкретної роботи і оцінюється сумою набраних балів:

для дисциплін з формою семестрового контролю екзамен (іспит): максимальна сума – 60 балів; мінімальна сума, що дозволяє здобувачу вищої освіти скласти екзамен (іспит) – 35 балів.

**Підсумковий контроль** включає семестровий контроль та атестацію здобувача вищої освіти.

**Семестровий контроль** проводиться у формах семестрового екзамену (іспиту), диференційованого заліку або заліку. Складання семестрового екзамену (іспиту) здійснюється під час екзаменаційної сесії.

Максимальна сума балів, яку може отримати здобувач вищої освіти під час екзамену (іспиту) – 40 балів. Мінімальна сума, за якою екзамен (іспит) вважається складеним – 25 балів.

**Підсумкова оцінка за навчальною дисципліною** визначається: для дисциплін з формою семестрового контролю екзамен (іспит) – сумуванням балів за поточний та підсумковий контроль.

Під час викладання навчальної дисципліни використовуються наступні контрольні заходи:

Поточний контроль: виконання лабораторних робіт та їх захист (40 балів), виконання тестових завдань (20 балів), екзамен – 40 балів.

Семестровий контроль: Екзамен.

Більш детальну інформацію щодо системи оцінювання наведено в робочому плані (технологічній карті) з навчальної дисципліни.

Приклад екзаменаційного білета та критерії оцінювання для навчальної дисципліни з формою семестрового контролю екзамен (іспит).

### Приклад екзаменаційного білета

Харківський національний економічний університет імені Семена Кузнеця  
Перший (бакалаврський) рівень вищої освіти  
Спеціальність "Інформаційні системи і технології"  
Освітньо-професійна програма "Штучний інтелект".  
Навчальна дисципліна "Основи алгоритмізації"

### ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 1

#### Завдання 1 (тестове) (20 балів).

Виконайте тестове завдання на сайті ПНС.

#### Завдання 2 (стереотипне) (10 балів).

Розробіть оптимальний алгоритм та реалізуйте його для розв'язку задачі, вказаної нижче. Додайте власні коментарі, що детально пояснюють код. Виконайте тестування роботи програми на різних даних.

В програму вводяться 2 числа, які описують час: кількість годин (від 0 до 23) та кількість хвилин (від 0 до 59). Надрукуйте текстове повідомлення, яке відповідає часу доби: від 22:00 до 6:59 – ніч, від 7:00 до 17:59 – день, від 18:00 до 21:59 – вечір. Надрукуйте відповідне повідомлення, якщо введено некоректні значення.

#### Завдання 3 (діагностичне) (10 балів).

Розробіть оптимальний алгоритм та реалізуйте його для розв'язку задачі, вказаної нижче. Додайте власні коментарі, що детально пояснюють код. Виконайте тестування роботи програми на різних даних. Вкажіть, яка алгоритмічна складність розробленого алгоритму та поясніть.

В програму вводяться два масиви з десяти значень кожен: список міст, які відвідав під час подорожі перший працівник, та список міст, які відвідав другий працівник. Надрукуйте загальний список всіх відвіданих міст без повторень (наприклад, якщо обидва працівники відвідали Київ – його треба надрукувати один раз).

Затверджено на засіданні кафедри інформатики та комп'ютерної техніки  
протокол № \_\_\_\_\_ від " \_\_\_\_ " \_\_\_\_\_ 20\_\_ р.

Екзаменатор

к.т.н., доц. Гороховатський О.В.

Зав. кафедрою

д.т.н., проф. Удовенко С.Г.

### Критерії оцінювання

Підсумкові бали за екзамен складаються із суми балів за виконання всіх завдань, округлені до цілого. За часткове виконання завдання нараховуються часткові бали.

### **Завдання 1 (тестове) (20 балів).**

За виконання складових завдання можна отримати наступні бали:

1 бал за кожне правильне тестове питання;

0 балів, якщо відповідь на тестове питання неправильна.

### **Завдання 2 (стереотипне) (10 балів).**

За виконання кожного складового завдання можна отримати по 2 бали:

Правильність:

2 бали, якщо алгоритм є правильним;

1 бал, якщо алгоритм є правильним, але містить незначні недоліки;

0 балів, якщо алгоритм є неправильним.

Оптимальність:

2 бали, якщо алгоритм є оптимальним;

1 бал, якщо алгоритм є оптимальним, але не найгіршим;

0 балів за оптимальність, якщо обрано найгірше рішення.

Тестування:

2 бали, якщо алгоритм протестовано на всіх можливих підмножинах рішень;

1 бал, якщо алгоритм протестовано не на всіх можливих підмножинах рішень;

0 балів, якщо тестування реалізації алгоритму не виконано.

Правильність на різних даних:

2 бали, якщо програма працює правильно для всіх даних;

1 бал, якщо програма не працює правильно для певних даних;

0 балів, якщо програма не працює, або працює неправильно.

Коментарі:

2 бали, якщо наявні детальні авторські коментарі опису алгоритму та коду;

1 бал, якщо детальні авторські коментарі опису алгоритму та коду наявні в недостатньому обсязі;

0 балів, якщо коментарі відсутні або неправильні.

### **Завдання 3 (діагностичне) (10 балів).**

За виконання кожного складового завдання можна отримати по 2 бали:

Правильність:

2 бали, якщо алгоритм є правильним;

1 бал, якщо алгоритм є правильним, але містить незначні недоліки;

0 балів, якщо алгоритм є неправильним.

Оптимальність:

2 бали, якщо алгоритм є оптимальним;

1 бал, якщо алгоритм є оптимальним, але не найгіршим;  
0 балів за оптимальність, якщо обрано найгірше рішення.

Тестування:

2 бали, якщо алгоритм протестовано на всіх можливих підмножинах рішень;  
1 бал, якщо алгоритм протестовано не на всіх можливих підмножинах рішень;  
0 балів, якщо тестування реалізації алгоритму не виконано.

Правильність на різних даних:

2 бали, якщо програма працює правильно для всіх даних;  
1 бал, якщо програма не працює правильно для певних даних;  
0 балів, якщо програма не працює, або працює неправильно.

Коментарі, складність:

2 бали, якщо наявні детальні авторські коментарі опису алгоритму та коду, наявне правильне обґрунтування алгоритмічної складності;  
1 бал, якщо детальні авторські коментарі опису алгоритму та коду наявні в недостатньому обсязі або відсутнє правильне обґрунтування алгоритмічної складності;  
0 балів, якщо коментарі відсутні або неправильні та відсутнє правильне обґрунтування алгоритмічної складності.

## РЕКОМЕНДОВАНА ЛІТЕРАТУРА

### Основна

1. Алгоритмізація та програмування: Практикум [Електронний ресурс]: навч. посіб. для здобувачів ступеня бакалавра за спеціальністю 122 "Комп'ютерні науки" / Л. І. Кублій; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 28,15 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2019. – 209 с
2. Розробка та аналіз алгоритмів [Електронний ресурс]: навчальний посібник / Г. В. Солодовник, О. О. Шаповалова; Харківський національний економічний університет ім. С. Кузнеця. – Електрон. текстові дан. (4,35 МБ). – Харків : ХНЕУ ім. С. Кузнеця, 2023. – 250 с. – Загол. з титул. екрану. – Бібліогр.: с. 223-224. URL: <http://repository.hneu.edu.ua/handle/123456789/32469>
3. Фратавчан В.Г., Фратавчан Т.М., Лазорик В.В. Алгоритмізація та програмування, навчальний посібник для закладів вищої освіти. – ЧНУ, 2022, – 286 с.

### Додаткова

4. Основи алгоритмізації. Методичні рекомендації до виконання лабораторних робіт для студентів спеціальності 122 "Комп'ютерні науки" освітньої програми "Комп'ютерні науки" першого (бакалаврського) рівня [Електронний ресурс] / уклад. О. В. Щербаков, О. В. Фролов; Харківський національний економічний університет ім. С. Кузнеця. – Електрон. текстові дан. (3,82 МБ). – Харків : ХНЕУ ім. С. Кузнеця, 2023. – 68 с. : іл. – Загол. з титул.

екрану – Бібліогр.: с. 66-67. URL:  
<http://repository.hneu.edu.ua/handle/123456789/29585>

5. Трофименко О. Г., Прокоп Ю. В., Задерейко О. В. Алгоритмізація та програмування : навчально-методичний посібник. – Одеса : Фенікс, 2020. 310 с. URL: <http://dspace.onua.edu.ua/handle/11300/12345>.

6. R. Mantaci, J. Yun. Basics of Programming and Algorithms, Principles and Applications. – Birkhäuser Cham, 2024, 978-3-031-59800-5. – 362 p. – Режим доступу до ресурсу: <https://link.springer.com/book/10.1007/978-3-031-59801-2>

### **Інформаційні ресурси**

7. Основи алгоритмізації (6.04.F4.030.25.1, 6.04.F6.020.25.1, 6.04.F6.010.25.1, доц. О. Гороховатський). – Режим доступу до ресурсу: <https://pns.hneu.edu.ua/course/view.php?id=8831>.

8. С# Підручник. – Режим доступу до ресурсу: <https://w3schoolsua.github.io/cs/index.html#gsc.tab=0>

9. Teachers2IT: Основи мови програмування С#. – Режим доступу до ресурсу: <https://www.youtube.com/watch?v=Z3YhrKhgO7M>

10. С# Tutorial For Beginners & Basics. – Режим доступу до ресурсу: [https://www.youtube.com/watch?v=67oWw9TanOk&list=PL82C6-O4XrHfoN\\_Y4MwGvJz5BntiL0z0D](https://www.youtube.com/watch?v=67oWw9TanOk&list=PL82C6-O4XrHfoN_Y4MwGvJz5BntiL0z0D)