



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
МІЖНАРОДНИЙ ГУМАНІТАРНИЙ УНІВЕРСИТЕТ
Кафедра інформаційних технологій

«ЗАТВЕРДЖУЮ»



Ректор Міжнародного гуманітарного
університету Д.Ю.Н., професор

Костянтин ГРОМОВЕНКО

_____ 2025 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ

Галузь знань	<u>01 «Освіта/Педагогіка»</u>
Спеціальність	<u>014 «Середня освіта»</u>
Назва освітньої програми	<u>Інформатика та програмування</u>
Рівень вищої освіти	<u>перший (бакалаврський)</u>

Одеса - 2025 рік

Робоча програма затверджена на засіданні кафедри комп'ютерних наук.

Протокол № 1 від 28.08 2025 року.

Розробники і викладачі	Контактний тел.	E-mail
професор кафедри інформаційних технологій, доктор технічних наук, професор Приходько Сергій Борисович	+380993876131	sergiy.prykhodko@nuos.edu.ua

Завідувач кафедри інформаційних
технологій к.т.н., доцент



Тетяна ГРИГОР'ЄВА

Гарант освітньої програми
к.т.н., доцент



Віктор ГОРБАЧОВ

Узгоджено
Начальник навчального відділу



Лілія САЄНКО

1. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни	
Кількість кредитів – 6, загальна кількість годин – 180	Галузь – <u>01 «Освіта/Педагогіка»</u> Спеціальність – <u>014.09 «Середня освіта (Інформатика)»</u>	Денна форма	Заочна форма
		Обов'язкова	
		Рік підготовки:	
		4	
		Семестр:	
Мова навчання – українська	Рівень вищої освіти – перший (бакалаврський)	8	
		Лекційні заняття:	
		40	12
		Практичні заняття:	
		40	12
		Лабораторні роботи	
		0	0
		Самостійна робота	
		100	156
		Вид контролю:	
		екзамен	екзамен

1. АНОТАЦІЯ ДО КУРСУ

МЕТА ДИСЦИПЛІНИ. Отримання студентами базових знань і практичних навиків основ сучасної технології розробки і реалізації складних і програмних продуктів. Отримані знання можуть бути використані в розробці прикладних програм.

Завдання – підготовка фахівців, що володіють сучасними методами і засобами розробки алгоритмів та програм, що знають сучасну технологію програмування і уміють застосовувати її при рішенні складних прикладних задач.

Ціль курсу – опанування стеком технологій об'єктно-орієнтованого програмування, його використання для розробки прикладного програмного забезпечення із використанням стандартних фреймворків. Курс передбачає теоретичні та практичні заняття, самостійне навчання та індивідуальні завдання.

ПРЕРЕКВІЗИТИ. Даний курс базується на знаннях та навичках, які отримані здобувачами при вивченні дисциплін «Алгоритмізація та програмування», «Алгоритми та структури даних», «Веб-програмування».

ПОСТРЕКВІЗИТИ. Знання, уміння та навички, набуті під час вивчення дисципліни, використовуються при подальшому вивченні дисциплін бакалаврської підготовки та практики.

2. ОЧІКУВАНІ КОМПЕТЕНТНОСТІ, ЯКІ ПЛАНУЄТЬСЯ СФОРМУВАТИ, ТА ДОСЯГНЕННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Інтегральна компетентність

ІК. Здатність розв'язувати складні спеціалізовані задачі у галузі освіти, інформатики та інформаційних технологій, що передбачає застосування певних теорій і методів педагогічних та комп'ютерних наук і характеризується комплексністю та невизначеністю умов.

Загальні компетентності

ЗК3. Здатність спілкуватися іноземною мовою.

ЗК8. Здатність приймати обґрунтовані рішення, генерувати нові ідеї, виявляти, ставити та вирішувати проблеми.

Фахові компетентності

ФК9. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.

ФК10. Здатність брати участь у проєктуванні програмного забезпечення, включаючи проведення моделювання (формального опису) його структури, поведінки та процесів функціонування, розробляти архітектури, модулі та компоненти програмних систем.

Фахові програмні результати навчання

РН3. Спілкуватися іноземною мовою у професійній діяльності.

Предметні програмні результати навчання

ПРН2. Створювати цифрові продукти, зокрема програми, для розв'язання задач/проблем за допомогою цифрових пристроїв.

ПРН5. Розв'язувати задачі з інформатики різного рівня складності раціональними методами.

ПРН7. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

Заплановані результати навчання за навчальною дисципліною

Знання:

Студент повинен знати:

- основні етапи розробки програмного забезпечення та процеси його життєвого циклу;
- поняття алгоритму, його властивості та способи подання алгоритмів;
- базові структури алгоритмів (лінійні, розгалужені, циклічні);
- методи оцінювання ефективності та складності алгоритмів;
- алфавіт та синтаксис мови програмування C++;
- основні типи даних, правила оголошення змінних і констант;
- правила запису арифметичних виразів та використання математичних функцій;
- оператори введення та виведення даних у C++;
- оператори керування програмним потоком: умовні оператори, оператори вибору, оператори циклів;
- принципи організації функцій у програмі;
- відмінності між локальними та глобальними змінними;
- способи передавання параметрів у функції;
- принципи використання масивів у програмуванні;
- методи опрацювання одновимірних масивів.

Уміння:

- аналізувати поставлену задачу та формалізувати її у вигляді алгоритму;
- складати алгоритми розв'язування обчислювальних задач різної складності;
- реалізовувати алгоритми мовою програмування C++;
- використовувати базові структури алгоритмів при створенні програм;
- створювати програми з використанням умовних операторів та операторів вибору;
- програмувати циклічні процеси та вкладені цикли;
- виконувати обчислення сум, добутків та табулювання функцій;
- працювати з функціями: оголошувати їх, передавати параметри та повертати результати;

- використовувати локальні та глобальні змінні;
- розробляти програми з використанням одновимірних масивів;
- реалізовувати алгоритми обробки масивів (пошук, сортування, підрахунок елементів тощо);
- організовувати програмний код у вигляді функціональних модулів.

Навички:

- алгоритмічного мислення та формалізації задач;
- розробки, налагодження та тестування програм мовою C++;
- використання сучасних підходів до структурного та модульного програмування;
- ефективного використання операторів керування програмним потоком;
- роботи з функціями та параметрами різних типів;
- обробки числових даних та масивів;
- оптимізації алгоритмів з урахуванням їх складності;
- написання структурованого та читабельного програмного коду;
- пошуку та виправлення помилок у програмному коді.

3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Тема 1. Концепція ООП. Основні поняття ООП. Правила іменування Java.

Принципи об'єктно-орієнтованого підходу як основи проєктування програмних систем.

Формування коректної термінології (англійською мовою). Аналіз підходів до іменування як елементу читабельності та підтримки коду.

Тема 2. Об'єкти та класи. Об'єкт у Java. Клас у Java. Поле в Java. Метод у Java. Анонімний об'єкт.

Моделювання предметної області через класи та об'єкти, побудови інформаційних моделей даних.

Розгляд підходів до структурування програм і створення базових програмних компонентів.

Тема 3. Метод у Java. Заздалегідь визначений метод. Метод, визначений користувачем. Статичний метод. Метод екземпляра. Абстрактний метод.

Підходи до декомпозиції задач і повторного використання коду через методи. Вибір типу методу залежно від задачі та логіки програмної системи.

Тема 4. Конструктори. Конструктор Java за замовчуванням. Перевантаження конструктора в Java.

Відмінність між конструктором та методом у Java. Конструктор копіювання у Java. Копіювання значень без конструктора.

Механізми ініціалізації об'єктів та проєктування життєвого циклу об'єкта. Аналіз способів створення гнучких і безпечних структур даних.

Тема 5. Ключове слово static у Java. Статичне поле в Java. Статичний метод у Java. Статичний блок у Java.

Особливості роботи зі статичними елементами як засобами оптимізації та організації спільних ресурсів. Аналіз архітектурних рішень у програмуванні.

Тема 6. Ключове слово this у Java. Посилання на змінну екземпляра поточного класу. Виклик методу поточного класу. Виклик конструктора поточного класу.

Керування контекстом об'єкта та зв'язками між його компонентами. Приклади коректного використання внутрішніх посилань у складних програмних структурах.

Тема 7. Інкапсуляція в Java.

Принципи приховування даних і забезпечення безпеки програмних компонентів. Прийоми проєктування надійних і підтримуваних систем.

Тема 8. Пакети. Пакет у Java. Підпакет у Java.

Засоби організації програмного коду у великомасштабних проєктах. Методи структуризації програм і роботи з багатомодульними системами.

Тема 9. Успадкування в Java. Однорівневе успадкування. Багаторівневе успадкування. Ієрархічне успадкування.

Механізми повторного використання коду та побудови ієрархій класів. Аналіз підходів до розширення функціональності програмних систем.

Тема 10. Відносини між класами в Java. IS-A відносини. HAS-A відносини.

Прийоми моделювання зв'язків між об'єктами в предметній області. Вибір оптимальних архітектурних рішень при проєктуванні програм.

Тема 11. Перевантаження методу в Java. Перевантаження методу: зміна кількості аргументів.

Перевантаження методу: зміна типу даних аргументів. Метод перевантаження та просування типу.

Способи реалізації поліморфної поведінки на етапі компіляції. Приклади створення гнучких інтерфейсів програмних компонентів.

Тема 12. Перевизначення методу в Java.

Механізми динамічного поліморфізму та перевизначення поведінки об'єктів. Методи адаптації програм до змін вимог.

Тема 13. Поліморфізм у Java. Поліморфізм під час виконання в Java. Модернізація. Приклад поліморфізму під час виконання в Java.

Універсальність обробки об'єктів різних типів та побудова розширюваних систем. Приклади ефективного застосування поліморфізму при розв'язанні задач.

Тема 14. Абстрактний клас у Java. Абстракція в Java. Абстрактний клас. Абстрактний метод.

Абстрактний клас, що містить конструктор, поля та методи.

Принципи абстрагування як основи моделювання складних систем. Створення узагальнених архітектур і шаблонів програмування.

4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви змістових модулів і тем	Кількість годин							
	денна форма				Заочна форма			
	усього	у тому числі			усього	у тому числі		
		лекц.	пракс.	сам.роб.		лекц.	пракс.	сам.роб.
Тема 1. Концепція ООП. Основні поняття ООП. Правила іменування Java.	12	4	4	4	12	2		10
Тема 2. Об'єкти та класи. Об'єкт у Java. Клас у Java. Поле в Java. Метод у Java. Анонімний об'єкт.	12	2	2	8	12	-	2	10
Тема 3. Метод у Java. Заздалегідь визначений метод. Метод, визначений користувачем. Статичний метод. Метод екземпляра. Абстрактний метод.	12	4	4	4	12	2	2	8
Тема 4. Конструктори. Конструктор Java за замовчуванням. Перевантаження конструктора в Java. Відмінність між конструктором та методом у Java. Конструктор копіювання у Java. Копіювання значень без конструктора.	12	2	2	8	12	2	2	8
Тема 5. Ключове слово static у Java. Статичне поле в Java. Статичний метод у Java. Статичний блок у Java.	12	4	4	4	12	-	2	10
Тема 6. Ключове слово this у Java. Посилання на змінну екземпляра поточного класу. Виклик методу поточного класу. Виклик конструктора поточного класу.	12	4	4	4	12	-	2	10
Тема 7. Інкапсуляція в Java.	12	2	2	8	12	2		10
Тема 8. Пакет у Java. Пакет у Java. Підпакет у Java.	12	4	4	4	12	2		10
Тема 9. Успадкування в Java. Однорівневе успадкування. Багаторівневе успадкування. Ієрархічне успадкування.	12	2	2	8	12	-	-	12
Тема 10. Відносини між класами в Java. IS-A відносини. HAS-A відносини.	12	4	4	4	12	-	2	10

Тема 11. Перевантаження методу в Java. Перевантаження методу: зміна кількості аргументів. Перевантаження методу: зміна типу даних аргументів. Метод перевантаження та просування типу.	15	4	4	7	15	2	-	13
Тема 12. Перевизначення методу в Java.	15	2	2	13	15			15
Тема 13. Поліморфізм у Java. Поліморфізм під час виконання в Java. Модернізація. Приклад поліморфізму під час виконання в Java.	15	-	-	13	15	-	-	15
Тема 14. Абстрактний клас у Java. Абстракція в Java. Абстрактний клас. Абстрактний метод. Абстрактний клас, що містить конструктор, поля та методи.	15	2	2	11	15	-	-	15
Загалом	180	40	40	100	180	12	12	156
Підсумковий контроль – екзамен								

5. ПРАКТИЧНІ РОБОТИ

Назва теми	Кількість годин	
	Денна форма	Заочна форма
Тема 1. Реалізація основних понять об'єктно-орієнтованого програмування та правил іменування в Java. Практика створення простих програм із дотриманням стандартів кодування.	4	-
Тема 2. Створення та використання класів і об'єктів у Java. Опис полів і методів, робота з анонімними об'єктами.	2	2
Тема 3. Розробка та використання методів у Java. Створення користувацьких методів, використання статичних та нестатичних методів.	4	2
Тема 4. Використання конструкторів у Java. Реалізація перевантаження конструкторів і створення об'єктів різними способами.	2	2
Тема 5. Застосування ключового слова static у Java. Робота зі статичними полями, методами та блоками.	4	2
Тема 6. Використання ключового слова this у Java. Організація взаємодії між полями, методами та конструкторами класу.	4	2
Тема 7. Реалізація інкапсуляції в Java. Створення класів із прихованими полями та методами доступу.	2	-
Тема 8. Організація програм у пакети та підпакети в Java. Структурування проєктів.	4	-
Тема 9. Реалізація успадкування в Java. Побудова ієрархій класів різних типів.	2	-
Тема 10. Моделювання відносин між класами (IS-A, HAS-A) у Java. Проєктування структури програм.	4	2
Тема 11. Реалізація перевантаження методів у Java. Використання різних варіантів параметрів.	4	-
Тема 12. Реалізація перевизначення методів у Java. Використання динамічного поліморфізму.	2	-
Тема 13. Застосування поліморфізму в Java при розв'язанні задач. Побудова гнучких програмних рішень.	-	-
Тема 14. Використання абстрактних класів і методів у Java. Створення узагальнених моделей і програмних компонентів.	2	-
Загалом	40	12

6. САМОСТІЙНА РОБОТА

До самостійної роботи студентів відносяться наступні види діяльності:

1. Ознайомлення з науковою та навчальною літературою відповідно зазначених у програмі тем.
2. Опрацювання лекційного матеріалу.

3. Підготовка до практичних занять.
4. Консультації з викладачем протягом семестру.
5. Самостійне опрацювання окремих питань навчальної дисципліни.
6. Підготовка та виконання індивідуальних завдань у вигляді есе, рефератів тощо.
7. Підготовка до підсумкового контролю.

Тематика та питання до самостійної підготовки та індивідуальних завдань

Назва теми	Кількість годин	
	Денна форма	Заочна форма
Тема 1. Концепція ООП. Основні поняття ООП. Правила іменування Java.	4	10
Тема 2. Об'єкти та класи. Об'єкт у Java. Клас у Java. Поле в Java. Метод у Java. Анонімний об'єкт.	8	10
Тема 3. Метод у Java. Стандартні (вбудовані) методи. Метод, визначений користувачем. Статичний метод. Метод екземпляра. Абстрактний метод.	4	8
Тема 4. Конструктори. Конструктор Java за замовчуванням. Перевантаження конструктора в Java. Відмінність між конструктором та методом у Java. Імітація конструктора копіювання у Java. Копіювання значень без конструктора.	8	8
Тема 5. Ключове слово static у Java. Статичне поле в Java. Статичний метод у Java. Статичний блок у Java.	4	10
Тема 6. Програмування розгалужених алгоритмів. Елементи програмування. Лінійні алгоритми. Алгоритм розгалуження.	4	10
Тема 7. Програмування циклів. Елементи програмування. Циклічні алгоритми. Обробка списку.	8	10
Тема 8. Оператори циклу з попередньою і наступною умовами while, do while.	4	10
Тема 9. Правила організації функцій.	8	12
Тема 10. Поняття масиву. Елементи програмування. Обробка одновимірних масивів	4	10
Тема 11. Опрацювання одновимірних масивів у функціях. Розробка програм по роботі з файловим введенням-виведенням. Розробка програм по роботі з класами.	7	13
Тема 12. Перевизначення методів у Java.	13	15
Тема 13. Поліморфізм у Java. Поліморфізм під час виконання в Java (runtime polymorphism). Перевизначення методів як механізм поліморфізму. Приклад поліморфізму під час виконання в Java.	13	15
Тема 14. Абстрактний клас у Java. Абстракція в Java. Абстрактний клас. Абстрактний метод. Абстрактний клас, що містить конструктор, поля та методи.	11	15
Загалом	100	156

7. ВИДИ ТА МЕТОДИ КОНТРОЛЮ

Види контролю	Складові оцінювання
Поточний контроль , який здійснюється під час проведення практичних (лабораторних) занять, виконання самостійної роботи, індивідуальних завдань, дослідної робота здобувача тощо	50%
Підсумковий контроль , який здійснюється у ході проведення екзамену	50%

Методи діагностики знань (контролю)	Фронтальне опитування, розв'язання практичних завдань, тестування здобувачів, наукові доповіді, реферати, усні повідомлення, екзамен
--	--

Питання для підсумкового контролю

1. Поняття програмного забезпечення та основні етапи його розробки.
2. Життєвий цикл програмного забезпечення та його основні процеси.
3. Поняття алгоритму та його основні властивості.
4. Основні способи подання алгоритмів.
5. Основні структури алгоритмів: лінійна, розгалужена, циклічна.
6. Методи оцінювання ефективності та складності алгоритмів.
7. Алфавіт та основні елементи мови програмування Java.
8. Основні типи даних у Java.
9. Змінні та константи у Java, правила їх оголошення.
10. Арифметичні операції та правила запису виразів у Java.
11. Оператори присвоювання та приведення типів у Java.
12. Стандартні математичні функції класу Math у Java.
13. Організація введення та виведення даних у Java.
14. Лінійні алгоритми та їх реалізація мовою Java.
15. Операції відношення та логічні операції у Java.
16. Умовний оператор if та його використання.
17. Тернарний умовний оператор у Java.
18. Оператор вибору switch та особливості його використання.
19. Поняття безумовного переходу та причини відсутності оператора goto у Java.
20. Цикл for та його використання у програмах.
21. Цикли while і do-while та їх особливості.
22. Використання вкладених циклів у програмах.
23. Алгоритми обчислення сум та добутків.
24. Табулювання функцій у програмуванні.
25. Опрацювання послідовностей чисел у програмах.
26. Обчислення сум із заданою точністю.
27. Поняття методу у Java та його структура.
28. Параметри методів: формальні та фактичні.
29. Передавання значень у методи та оператор return.
30. Область видимості змінних (локальні змінні та поля класу).
31. Поняття класу та об'єкта у Java.
32. Створення класів і об'єктів. Конструктори.
33. Інкапсуляція як принцип об'єктно-орієнтованого програмування.
34. Модифікатори доступу у Java.
35. Статичні та нестатичні члени класу.
36. Поняття масиву. Одновимірні масиви та їх обробка.
37. Двовимірні масиви та основні алгоритми їх опрацювання.
38. Передавання масивів у методи.
39. Пошук і сортування елементів масиву.
40. Обробка рядків у Java (клас String).
41. Спадкування класів у Java.
42. Поліморфізм та його реалізація у Java.
43. Перевизначення методів (override) та перевантаження (overload).
44. Абстрактні класи та інтерфейси.
45. Обробка винятків (exception handling) у Java.
46. Основи роботи з колекціями (ArrayList, List).
47. Робота з файлами у Java (основні класи та операції).
48. Розробка простої програми з використанням класів і методів.
49. Налаштування програм та пошук помилок.
50. Принципи об'єктно-орієнтованого програмування та їх застосування у Java..

8. ОЦІНЮВАННЯ ПОТОЧНОЇ, САМОСТІЙНОЇ ТА ІНДИВІДУАЛЬНОЇ РОБОТИ ЗДОБУВАЧІВ

Види роботи	Планові терміни виконання	Форми контролю та звітності	Максимальний відсоток оцінювання
Систематичність і активність роботи на практичних заняттях			
1.1. Підготовка до практичних занять	Відповідно до робочої програми та розкладу занять	Перевірка обсягу та якості засвоєного матеріалу під час практичних занять	30
Виконання завдань для самостійного опрацювання			
1.2. Індивідуальне тестування	Відповідно до робочої програми та розкладу занять	Перевірка результатів індивідуального тестування, перевірка конспектів лекцій, перевірка рефератів	10
Виконання індивідуальних завдань (дослідна робота здобувача)			
1.3. Інші види індивідуальних завдань, в т.ч. підготовка наукових публікацій, участь у роботі круглих столів, конференцій тощо.	Відповідно до робочої програми та розкладу занять	Обговорення результатів проведеної роботи під час аудиторних занять, наукових конференцій та круглих столів.	10
Разом балів за поточний контроль			50
Підсумковий контроль – екзамен			50
Загалом балів			100

Поточний контроль знань здобувачів при підсумковому контролі у формі екзамену (максимум 50 балів) оцінюється за шкалою («2», «3», «4», «5»), з обов'язковим переведенням балів за кожний вид роботи таким чином:

- за підготовку до аудиторних занять (максимум 30 балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості практичних (лабораторних) занять, для переводу балів за підготовку до аудиторних занять середньоарифметична оцінка множиться на коефіцієнт 6;
- за виконання завдань для самостійного опрацювання (тестування, завдання, підготовка реферату за заданою тематикою) (максимум 10 балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості програмного матеріалу, що виноситься на самостійне вивчення, для переводу зазначених балів середньоарифметична оцінка множиться на коефіцієнт 2;
- за інші види індивідуальних завдань, в т.ч. підготовку наукових публікацій, участь у роботі круглих столів, конференцій тощо (максимум 10 балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості інших видів індивідуальних завдань, для переводу балів за підготовку до аудиторних занять середньоарифметична оцінка множиться на коефіцієнт 2.

Критерії оцінювання поточного контролю

- «2» – виставляється за неправильну відповідь (письмову роботу), яка не відповідає змісту теми та свідчить про нерозуміння її основних положень;
- «3» – виставляється за відповідь (письмову роботу), яка відповідає змісту теми, але є неповною і неточною у визначенні понять, свідчить про неповне розуміння основних положень

навчального матеріалу, свідчить про те, що знання здобувача мають фрагментарний, поверховий характер;

– «4» – оцінюється правильна і обґрунтована відповідь (письмова робота), з якої видно, що здобувач знає й розуміє теоретичний матеріал в обсязі навчальної теми, володіє необхідними навичками, не допускає суттєвих помилок.

– «5» – оцінюється повна, правильна, ґрунтовна відповідь (письмова робота) на запитання теми, що передбачає логічність і послідовність викладу матеріалу, володіння термінологією, показує вміння повно й глибоко використовувати теоретичні знання в практичній площині.

Критерії оцінювання підсумкового контролю у формі екзамену

Максимальна оцінка підсумкового контролю у формі екзамену не може перевищувати 50 балів. При цьому рівень знань оцінюється:

– **від 40 до 50 балів** – здобувач виявляє особливі творчі здібності, вміє самостійно знаходити та опрацьовувати необхідну інформацію, демонструє знання матеріалу, проводить узагальнення і висновки;

– **від 30 до 40 балів** – здобувач володіє знаннями матеріалу, але допускає незначні помилки у формуванні термінів, категорій, проте за допомогою викладача швидко орієнтується і знаходить правильні відповіді;

– **від 20 до 30 балів** – здобувач відтворює значну частину теоретичного матеріалу, виявляє знання і розуміння основних положень, з допомогою викладача може аналізувати навчальний матеріал, але дає недостатньо обґрунтовані, невичерпні відповіді, допускає помилки.

– **від 10 до 20 балів** – здобувач володіє навчальним матеріалом на середньому рівні, допускає помилки, серед яких є значна кількість суттєвих;

– **від 0 до 10 балів** – здобувач володіє навчальним матеріалом на рівні, вищому за початковий, значну частину його відтворює на репродуктивному рівні, на всі запитання дає необґрунтовані, невичерпні відповіді, допускає помилки.

9. ОЦІНЮВАННЯ ЗАГАЛЬНИХ РЕЗУЛЬТАТІВ ЗНАНЬ ЗДОБУВАЧІВ

Загальні результати знань здобувачів по даній дисципліні оцінюються наступним чином:

– **«Відмінно»/«зараховано» (А)** – від 90 до 100 балів. Здобувач виявляє особливі творчі здібності, вміє самостійно знаходити та опрацьовувати необхідну інформацію, демонструє знання матеріалу, проводить узагальнення і висновки. Був присутній на лекціях та семінарських заняттях, під час яких давав вичерпні, обґрунтовані, теоретично і практично правильні відповіді, має конспект з виконаними завданнями до самостійної роботи, презентував реферат за заданою тематикою, проявляє активність і творчість у науково-дослідній роботі;

– **«Добре»/«зараховано» (В)** – від 82 до 89 балів. Здобувач володіє знаннями матеріалу, але допускає незначні помилки у формуванні термінів, категорій, проте за допомогою викладача швидко орієнтується і знаходить правильні відповіді. Був присутній на лекціях та семінарських заняттях, має конспект з виконаними завданнями до самостійної роботи, презентував реферат за заданою тематикою, проявляє активність і творчість у науково-дослідній роботі;

– **«Добре»/«зараховано» (С)** – від 74 до 81 балів. Здобувач відтворює значну частину теоретичного матеріалу, виявляє знання і розуміння основних положень, з допомогою викладача може аналізувати навчальний матеріал, але дає недостатньо обґрунтовані, невичерпні відповіді, допускає помилки. При цьому враховується наявність конспекту з виконаними завданнями до самостійної роботи, реферату та активність у науково-дослідній роботі;

– **«Задовільно»/«зараховано» (D)** – від 64 до 73 балів. Здобувач був присутній не на всіх лекціях та семінарських заняттях, володіє навчальним матеріалом на середньому рівні, допускає помилки, серед яких є значна кількість суттєвих. При цьому враховується наявність конспекту з виконаними завданнями до самостійної роботи та рефератів;

– **«Задовільно»/«зараховано» (Е)** – від 60 до 63 балів. Здобувач був присутній не на всіх лекціях та семінарських заняттях, володіє навчальним матеріалом на рівні, вищому за початковий,

значну частину його відтворює на репродуктивному рівні, на всі запитання дає необґрунтовані, невичерпні відповіді, допускає помилки, має неповний конспект з завданнями до самостійної роботи.

– «Незадовільно з можливістю повторного складання»/«не зараховано» (FX) – від 35 до 59 балів. Здобувач володіє матеріалом на рівні окремих фрагментів, що становлять незначну частину навчального матеріалу.

– «Незадовільно з обов’язковим повторним вивченням дисципліни»/«не зараховано» (F) – від 0 до 34 балів. Здобувач не володіє навчальним матеріалом.

Таблиця відповідності результатів контролю знань за різними шкалами

100-бальна шкала	Шкала за ECTS	Національна шкала	
		Екзамен	Залік
90-100	A	Відмінно	Зараховано
82-89	B	Добре	Зараховано
74-81	C		
64-73	D	Задовільно	Зараховано
60-63	E		
35-59	FX	Незадовільно	Не зараховано
1-34	F		

10. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Об’єктно-орієнтоване програмування мовою Java [Електронний ресурс]: навч. посіб. для здобувачів ступеня бакалавра / КПІ ім. Ігоря Сікорського; уклад.: Д. Д. Татарчук, Ю. В. Діденко, Г. С. Свєчніков. – Київ: КПІ ім. Ігоря Сікорського, 2024. – 153 с. URL: <https://ela.kpi.ua/server/api/core/bitstreams/51fa06c7-4b06-4138-a35f-a5e57ef15e30/content>
2. Об’єктно-орієнтоване програмування мовою Java. [Електронний ресурс]: навч. посіб. / уклад.: О.М. Ткаченко, В.А. Каплун – Вінниця: ВНТУ, 2006. – 107 с. URL: <https://ot.vntu.edu.ua/wp-content/uploads/2013/02/oop.pdf>
3. Горбань А.В. Програмування в Java: Навчальний посібник [електронний ресурс]. –2008. – 310 с. URL: <http://programming.in.ua/programming/basisprogramming/144-programming-Java-book.html>
4. Брнакевич І.Є., Вагін П.П. Програмування мовою Java: використання фундаментальних класів: Тексти лекцій. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2002. – 75 с. URL: http://blues.franko.lviv.ua/ami/books/ami/Java_fundamental.pdf
5. Kathy Sierra, Bert Bates. "Head First Java". O'Reilly Media, 2005. - 688 p.
6. Joshua Bloch. "Effective Java". Addison-Wesley, 2018. - 416 p.
7. Herbert Schildt. "Java: The Complete Reference". McGraw-Hill Education, 2018. - 1248 p.
8. Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes, Doug Lea. "Java Concurrency in Practice". Addison-Wesley, 2006. - 424 p.
9. Robert C. Martin. "Clean Code: A Handbook of Agile Software Craftsmanship". Prentice Hall, 2008. - 464 p.
10. Scott Oaks. "Java Performance: The Definitive Guide". O'Reilly Media, 2014. - 425 p.
11. Craig Walls. "Spring in Action". Manning Publications, 2022. - 520 p.

12. Bart Baesens, Aimee Backiel, Seppe vanden Broucke. "Beginning Java Programming: The Object-Oriented Approach". Wiley, 2015. - 672 p.

Допоміжна

13. Prykhodko S.B. Nonlinear regression models for estimating the duration of software development in Java for PC based on the 2021 ISBSG data / S. B. Prykhodko, A. V. Pukhalevych, K. S. Prykhodko, L. M. Makarova // Radio Electronics, Computer Science, Control. – No. 3 (62). – 2022. – P. 144-154. (Web of Science) DOI: <https://doi.org/10.15588/1607-3274-2022-3-14>
14. Prykhodko S. A Statistical Evaluation of the Depth of Inheritance Tree Metric for Open-Source Applications Developed in Java / S. Prykhodko, N. Prykhodko, T. Smykodub // Foundations of Computing and Decision Sciences. – 2021. – Vol. 46. – No. 2. – P. 159-172. (Scopus, Web of Science) <https://doi.org/10.2478/fcds-2021-0011>
15. Prykhodko S. A Joint Statistical Estimation of the RFC and CBO Metrics for Open-Source Applications Developed in Java / S. Prykhodko, N. Prykhodko, T. Smykodub // 2022 IEEE 17th International Conference on Computer Sciences and Information Technologies (CSIT), 2022. – P. 442-445, DOI: <https://doi.org/10.1109/CSIT56902.2022.10000457>
16. Васильєв О.М. Програмування мовою Java. Програмування мовою Java / О.М. Васильєв. Тернопіль : Навчальна книга. Богдан, 2019. 696 с.
17. Роберт Мартін. Чистий код: створення і рефакторинг за допомогою Agile / пер. з англ. І.Бондар-Терещенко. Харків : Видво «Ранок» Ж Фабула. 2021. 448 с.
18. Josh Juneau, Luciano Manelli. Java 17 Recipes: A Problem-Solution Approach. 2022 589 p. DOI: <https://doi.org/10.1007/978-1-4842-79632>
19. Mastering Java. A Beginner's Guide. Edited by Sufyan bin Uzayr. Taylor & Francis Group, LLC. 491 p. DOI: 10.1201/9781003229063
20. Michael Inden. Java Challenges: 100+ Proven Tasks that Will Prepare You for Anything. Apress, 2022. DOI: <https://doi.org/10.1007/978-14842-7395-1>

Інформаційні ресурси

21. <https://docs.oracle.com/en/java/>
22. <https://docs.oracle.com/javase/tutorial/>
23. <https://www.baeldung.com/>
24. <https://www.geeksforgeeks.org/java/>