



Міжнародний гуманітарний університет
Факультет кібербезпеки, програмної інженерії
та комп'ютерних наук
Кафедра комп'ютерної інженерії та інноваційних
технологій



«ЗАТВЕРДЖУЮ»

Ректор Міжнародного
гуманітарного університету

д.ю.н., професор

Костянтин ГРОМОВЕНКО

сергій 2025 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
СКРИПТОВІ МОВИ ПРОГРАМУВАННЯ

Галузь знань	<u>12 Інформаційні технології</u>
Спеціальність	<u>121 Інженерія програмного забезпечення</u>
Назва освітньої програми	<u>Інженерія програмного забезпечення</u>
Рівень вищої освіти	<u>перший (бакалаврський)</u>

Робоча програма затверджена на засіданні кафедри комп'ютерної інженерії та інноваційних технологій.

Протокол № 1 від 28.08. 2025 року.

Розробники і викладачі	Контактний тел.	E-mail
кандидат технічних наук, доцент Педяш Володимир Віталійович	+380673787003	vpedyash@gmail.com

Завідувачка кафедри комп'ютерної інженерії

та інноваційних технологій

к.т.н., доцент

 Лариса ЙОНА

Гарант освітньої програми

к.т.н., доцент

 Олександр РУСУ

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 – 12 Інформаційні технології Спеціальність – 121 Інженерія програмного забезпечення	Денна форма	Заочна форма
		вибіркова	
		Рік підготовки:	
		4	7
		Семестр:	
Мова навчання – українська	Рівень вищої освіти – перший (бакалаврський)	7	7
		Лекційні заняття:	
		40	12
		Практичні заняття:	
		38	12
		Лабораторні роботи	
		0	0
		Самостійна робота	
		102	156
		Вид контролю:	
		залік	

Дисципліна «Скриптові мови програмування» спрямована на формування у здобувачів вищої освіти спеціальності Інженерія програмного забезпечення компетентностей, пов'язаних із розробкою програмного забезпечення із використанням сучасних скриптових мов програмування. Вона орієнтована на формування здатності розробляти алгоритми, створювати програмні модулі та програмні системи для розв'язання прикладних задач різної складності, застосовувати стандартні та сторонні бібліотеки для обробки даних, автоматизації процесів та інтеграції програмних компонентів.

У межах дисципліни розглядаються основні конструкції скриптових мов програмування, зокрема робота зі змінними, типами даних, умовними операторами, циклами, функціями, модулями та класами. Особлива увага приділяється застосуванню структурованого та об'єктно-орієнтованого підходів до розробки програмного забезпечення, організації програмних проєктів, повторному використанню програмного коду, а також використанню бібліотек і програмних фреймворків для створення прикладних програм. У процесі навчання здобувачі освіти набувають практичних навичок використання інструментів налагодження, тестування та супроводу програмного забезпечення, що відповідає сучасним підходам до розробки програмних продуктів.

Актуальність дисципліни зумовлена широким застосуванням скриптових мов програмування у сучасній індустрії програмного забезпечення. Вони активно використовуються для розробки програмних систем, вебзастосунків, сервісів автоматизації, інструментів обробки даних, тестування програмного забезпечення та створення прототипів програмних продуктів. Простота синтаксису, кросплатформеність та наявність великої кількості відкритих бібліотек роблять скриптові мови програмування ефективним інструментом для розробника програмного забезпечення.

МЕТА ДИСЦИПЛІНИ. Мета дисципліни «Скриптові мови програмування» полягає у формуванні у здобувачів вищої освіти теоретичних знань та практичних навичок

використання скриптових мов програмування для розробки програмного забезпечення, створення алгоритмів, програмних модулів і програмних систем, а також застосування сучасних бібліотек і інструментів розробки у процесі створення програмних продуктів.

ПЕРЕКВІЗИТИ: Передумовами для вивчення дисципліни є знання та практичні навички, сформовані під час опанування навчальних дисциплін Алгоритмізація та програмування, Алгоритми та структури даних, Технології створення програмних продуктів, Об'єктно-орієнтоване програмування, Організація баз даних та знань. Ці дисципліни забезпечують базову підготовку з розробки алгоритмів, використання основних конструкцій програмування, принципів побудови програмних систем, роботи з даними та застосування об'єктно-орієнтованого підходу під час створення програмного забезпечення.

ПОСТРЕКВІЗИТИ: Програмування інфокомунікаційних сервісів і систем, Патерни та фреймворки, кваліфікаційна робота. Знання та вміння, отримані під час вивчення дисципліни, можуть використовуватися у подальшій професійній діяльності під час розробки програмного забезпечення, створення програмних сервісів, автоматизації процесів та інтеграції програмних систем.

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

У процесі реалізації програми дисципліни формуються наступні компетентності із передбачених освітньою програмою:

Інтегральна компетентність

Здатність розв'язувати складні спеціалізовані завдання або практичні проблеми інженерії програмного забезпечення, що характеризуються комплексністю та невизначеністю умов, із застосуванням теорій та методів інформаційних технологій

Загальні компетентності

K01. Здатність до абстрактного мислення, аналізу та синтезу.

K02. Здатність застосовувати знання у практичних ситуаціях.

K05. Здатність вчитися і оволодівати сучасними знаннями.

K06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

Спеціальні (фахові, предметні) компетентності

K15. Здатність розробляти архітектури, модулі та компоненти програмних систем.

K19. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.

K20. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.

K25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

K26. Здатність до алгоритмічного та логічного мислення.

K27. Здатність розробляти програмні модулі, додатки та скрипти для інтеграції, моніторингу та управління інфокомунікаційними системами, сервісами та пристроями з використанням програмних інтерфейсів і сучасних мов програмування.

Програмні результати навчання

ПР01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.

ПР07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

ПР13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

ПР14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проєктування, тестування, візуалізації, вимірювань та документування програмного забезпечення.

ПР15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.

ПР18. Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.

ПР25. Вміти розробляти програмні модулі, додатки та скрипти для інтеграції, моніторингу та управління інфокомунікаційними системами, сервісами та пристроями з використанням програмних інтерфейсів і сучасних мов програмування

Заплановані результати навчання за навчальною дисципліною

Знання:

- основні поняття та принципи використання скриптових мов програмування у процесі розробки програмного забезпечення;
- синтаксис сучасних скриптових мов програмування та правила оформлення програмного коду;
- основні типи даних (числові, логічні, рядкові, колекції даних) та операції над ними;
- механізми керування виконанням програм (умовні оператори, цикли, оператори керування виконанням);
- принципи створення та використання функцій, передачі параметрів і повернення результатів;
- принципи модульної побудови програмного забезпечення та організацію програм у вигляді модулів і пакетів;
- базові принципи об'єктно-орієнтованого програмування (класи, об'єкти, атрибути, методи, інкапсуляція);
- методи обробки виняткових ситуацій та налагодження програм;
- принципи використання стандартних програмних бібліотек та підключення сторонніх бібліотек;
- основні напрями застосування скриптових мов програмування у розробці програмного забезпечення, автоматизації процесів, обробці даних та створенні програмних сервісів.

Уміння:

- розробляти алгоритми розв'язання прикладних задач та реалізовувати їх засобами скриптових мов програмування;

- використовувати базові конструкції програмування (змінні, типи даних, умовні оператори, цикли) для створення програм;
- створювати та застосовувати функції, передавати параметри та обробляти результати їх виконання;
- розробляти програми з використанням принципів структурованого та об'єктно-орієнтованого програмування;
- створювати програмні модулі та використовувати стандартні й сторонні бібліотеки;
- здійснювати введення, обробку та виведення даних, працювати з файлами та зовнішніми джерелами даних;
- застосовувати механізми обробки винятків для підвищення надійності програмного забезпечення;
- налагоджувати та тестувати програмний код, аналізувати та виправляти помилки;
- використовувати скриптові мови програмування для автоматизації обчислень, обробки даних та розв'язання прикладних задач інженерії програмного забезпечення;
- документувати програмний код відповідно до сучасних вимог та стандартів розробки програмного забезпечення.

Навички:

- розробки алгоритмів та програмних модулів із використанням скриптових мов програмування;
- застосування сучасних середовищ розробки та інструментів налагодження програм;
- використання бібліотек і програмних модулів для реалізації прикладних програмних рішень;
- організації програмного коду у вигляді модулів, пакетів та компонентів програмної системи;
- тестування, налагодження та супроводження програмного забезпечення.

3. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Тема 1. Основи скриптових мов програмування

Поняття скриптових мов програмування. Класифікація скриптових мов за призначенням. Особливості інтерпретованих мов програмування. Синтаксис і структура програм. Правила оформлення програмного коду. Лексичні елементи мови програмування. Змінні та їх типи. Динамічна типізація даних. Вирази та операції над даними. Арифметичні, логічні та рядкові операції. Перетворення типів даних. Умовні оператори та їх використання. Реалізація розгалужених алгоритмів. Вкладені умовні конструкції. Тернарні оператори. Цикли та їх види. Цикли з передумовою та післяумовою. Ітерація по колекціях. Оператори керування виконанням циклів. Організація керування виконанням програм. Побудова алгоритмів із використанням базових конструкцій.

Тема 2. Функції, модулі та структуризація програмного коду

Поняття функції та її призначення. Оголошення та виклик функцій. Передача параметрів у функції. Параметри за значенням і за посиланням. Іменовані та позиційні параметри. Значення за замовчуванням. Повернення результатів виконання функцій. Області видимості змінних. Локальні та глобальні змінні. Рекурсія та її застосування. Базові та рекурсивні випадки. Порівняння рекурсії та ітерації. Організація програм у вигляді модулів.

Імпорт модулів. Простори імен. Створення власних модулів. Організація пакетів. Повторне використання програмного коду. Принципи модульності. Використання стандартних бібліотек. Підключення сторонніх бібліотек. Керування залежностями у проєкті.

Тема 3. Об'єктно-орієнтоване програмування та обробка даних

Поняття класу та об'єкта. Атрибути та методи класу. Ініціалізація об'єктів. Конструктори. Інкапсуляція та приховування даних. Методи доступу до атрибутів. Організація програмних компонентів. Створення та використання класів. Робота з екземплярами класів. Основи наслідування. Перевизначення методів. Поліморфізм у програмуванні. Робота з колекціями даних. Списки, кортежі, множини та словники. Основні операції над колекціями. Ітерація по структурах даних. Введення даних у програму. Виведення результатів. Робота з файлами. Зчитування та запис даних. Обробка текстових і числових даних. Основні методи обробки даних у прикладних задачах.

Тема 4. Розробка прикладних програм та автоматизація задач

Обробка виняткових ситуацій. Типи винятків. Конструкції обробки винятків. Генерація винятків. Забезпечення надійності програм. Налаштування програмного коду. Методи пошуку та виправлення помилок. Використання засобів відлагодження. Логування подій у програмі. Тестування програмного забезпечення. Методи тестування програм. Перевірка коректності результатів. Розробка прикладних програм. Структура програмного проєкту. Використання бібліотек і програмних інтерфейсів. Робота з API. Інтеграція програмних компонентів. Автоматизація задач за допомогою скриптів. Обробка даних у прикладних задачах. Створення утиліт для автоматизації процесів. Організація та документування програмного коду.

4. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назви змістових модулів і тем	Кількість годин									
	Денна форма					Заочна форма				
	Заг.	у тому числі				Заг.	у тому числі			
		Лек.	Прак.	Лаб.	Сам. роб.		Лек.	Прак.	Лаб.	Сам. роб.
Тема 1. Основи скриптових мов програмування \.	45	10	10	0	30	45	2	2	0	41
Тема 2. Функції, модулі та структуризація програмного коду.	45	10	10	0	25	45	4	4	0	37
Тема 3. Об'єктно-орієнтоване програмування та обробка даних	45	10	10	0	25	45	4	4	0	37
Тема 4. Розробка прикладних програм та автоматизація задач	45	10	8	0	27	45	2	2	0	41
Загалом	180	40	38	0	102	180	12	12	0	156
Підсумковий контроль – залік										

5. ПРАКТИЧНІ ЗАНЯТТЯ

Назва теми	Кількість годин	
	Денна форма	Заочна форма
Тема 1. Основи скриптових мов програмування 1. Реалізація базових алгоритмів із використанням змінних, типів даних та операцій 2. Розробка програм із використанням умовних операторів 3. Організація циклічних обчислень у скриптових мовах програмування 4. Обробка та перетворення даних у програмі 5. Побудова алгоритмів керування виконанням програм	10	2
Тема 2. Функції, модулі та структуризація програмного коду 1. Створення та використання функцій у програмі 2. Реалізація рекурсивних алгоритмів 3. Організація програмного коду у вигляді модулів 4. Використання стандартних бібліотек у програмуванні 5. Підключення та використання сторонніх бібліотек	10	4
Тема 3. Об'єктно-орієнтоване програмування та обробка даних 1. Створення класів та об'єктів у програмі 2. Реалізація інкапсуляції та методів доступу до даних 3. Використання колекцій даних для обробки інформації 4. Робота з файлами: зчитування та запис даних 5. Розробка програм для обробки даних	10	4
Тема 4. Розробка прикладних програм та автоматизація задач 1. Реалізація обробки винятків у програмі 2. Налаштування та тестування програмного коду 3. Розробка програмного модуля для автоматизації задач 4. Використання бібліотек та API у прикладних програмах 5. Створення скриптів для автоматизації обробки даних	8	2
Загалом	38	12

6. САМОСТІЙНА РОБОТА

До самостійної роботи здобувачів відносяться наступні види діяльності:

1. Опрацювання лекційного матеріалу.
2. Підготовка до практичних занять.
3. Консультації з викладачем впродовж семестру.
4. Ознайомлення з додатковою літературою відповідно до зазначених у програмі тем.
5. Самостійне опрацювання окремих питань навчальної дисципліни.
6. Підготовка індивідуальних завдань у вигляді презентацій або рефератів.
7. Підготовка до підсумкового контролю.

Назва теми	Кількість годин	
	Денна форма	Заочна форма
Тема 1. Основи скриптових мов програмування 1. Поняття та класифікація скриптових мов програмування. Особливості їх використання у розробці програмного забезпечення. 2. Синтаксис скриптових мов програмування та правила написання програмного коду. 3. Основні типи даних та операції над ними. Представлення та обробка даних у програмах. 4. Умовні оператори та їх використання для реалізації розгалужених алгоритмів. 5. Циклічні конструкції та їх застосування для організації повторюваних обчислень.	30	41
Тема 2. Функції, модулі та структуризація програмного коду 1. Поняття функції та її роль у структуризації програмного коду. 2. Передача параметрів у функції та повернення результатів виконання. 3. Рекурсивні функції та особливості їх використання у програмуванні. 4. Організація програмного коду у вигляді модулів і пакетів. 5. Використання стандартних бібліотек та підключення сторонніх програмних бібліотек.	25	37
Тема 3. Об'єктно-орієнтоване програмування та обробка даних 1. Основні поняття об'єктно-орієнтованого програмування: класи, об'єкти, атрибути та методи. 2. Інкапсуляція та організація структури програмного коду у вигляді програмних компонентів. 3. Використання колекцій даних для зберігання та обробки інформації у програмах. 4. Робота з файлами: зчитування, запис та обробка даних. 5. Основні методи обробки даних у прикладних програмних задачах.	25	37
Тема 4. Розробка прикладних програм та автоматизація задач 1. Обробка виняткових ситуацій у програмах та забезпечення надійності програмного забезпечення. 2. Методи налагодження програмного коду та пошук помилок у програмних системах. 3. Тестування програм та перевірка коректності їх роботи. 4. Використання бібліотек і програмних інтерфейсів для розширення функціональних можливостей програм. 5. Застосування скриптових мов програмування для автоматизації задач обробки даних та створення прикладних програм.	27	41
Загалом	102	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 роботи здобувача тощо		100%
Підсумковий контроль – залік		
Методи діагностики знань (контролю)	Фронтальне опитування; наукова доповідь, реферат, усне повідомлення, індивідуальне опитування; робота у групах; ділова гра, розв'язання ситуаційних завдань, кейсів, виконання практичних завдань, залік	

Питання для підсумкового контролю

1. Поняття скриптових мов програмування та їх особливості
2. Класифікація скриптових мов програмування
3. Відмінності інтерпретованих і компільованих мов
4. Синтаксис і структура програми у скриптових мовах
5. Основні правила оформлення програмного коду
6. Поняття змінної та її роль у програмі
7. Основні типи даних у скриптових мовах
8. Динамічна типізація даних
9. Арифметичні операції та їх використання
10. Логічні операції та умови
11. Рядкові дані та операції над ними
12. Перетворення типів даних
13. Умовні оператори та їх застосування
14. Вкладені умовні конструкції
15. Цикли та їх види
16. Організація циклів із лічильником та умовою
17. Оператори керування виконанням циклів
18. Поняття функції та її призначення
19. Оголошення та виклик функцій
20. Передача параметрів у функції
21. Повернення значень із функцій
22. Області видимості змінних
23. Локальні та глобальні змінні
24. Поняття рекурсії
25. Переваги та недоліки рекурсії
26. Поняття модуля у програмуванні
27. Імпорт модулів та організація простору імен
28. Організація програм у вигляді пакетів
29. Використання стандартних бібліотек
30. Підключення сторонніх бібліотек
31. Поняття класу та об'єкта
32. Атрибути та методи класу
33. Інкапсуляція у програмуванні

34. Основи наслідування
35. Поліморфізм у програмуванні
36. Колекції даних та їх види
37. Основні операції над колекціями
38. Робота з файлами у програмі
39. Обробка винятків та їх призначення
40. Налаштування та тестування програмного коду

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	60
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	20
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.	20
Разом балів за поточний контроль			100
Загальний результат оцінювання			100

Поточний контроль знань здобувачів освіти при формі контролю залік (максимум 100 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 60 балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості практичних (лабораторних) занять, для переведу балів за підготовку до аудиторних занять середньоарифметична оцінка множиться на коефіцієнт 12;

– за виконання завдань для самостійного опрацювання (тестування, завдання, підготовка реферату (есе) за заданою тематикою) (максимум 20 балів) здобувачу освіти необхідно мати не менше 1/2 оцінок від загальної кількості програмного матеріалу, що виноситься на самостійне вивчення, для переведу зазначених балів середньоарифметична оцінка множиться на коефіцієнт 4;

– за інші види індивідуальних завдань, в т.ч. підготовку наукових публікацій, участь у роботі круглих столів, конференцій тощо (максимум 20- 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 множить на коефіцієнт 4.

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.

9. ОЦІНЮВАННЯ ЗАГАЛЬНИХ РЕЗУЛЬТАТІВ ЗНАНЬ ЗДОБУВАЧІВ

Загальні результати знань здобувачів по 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 практичних 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 практичних 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 практичних 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» (E) – від 60 до 63 балів. Здобувач був присутній не на всіх лекціях та практичних 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
0-34	F		

10. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Педяш В. В., Йона Л. Г., Мазур Г. Д., Русу О. П. Python-програмування: методичні рекомендації для практичних та лабораторних занять [Електронне видання] / кафедра комп’ютерної інженерії та інноваційних технологій Міжнародного гуманітарного університету. Одеса, 2025. 137 с.

2. Костюченко А. О. Основи програмування мовою Python: навчальний посібник. Черкаси: ФОП Баликіна С. М., 2020. 180 с.

3. Яковенко А. В. Основи програмування. Python. Частина 1: підручник для студентів спеціальності 122 Комп’ютерні науки, спеціалізації Інформаційні технології в біології та медицині. Київ: КПІ ім. Ігоря Сікорського, 2018. 195 с.

4. Маттес Е. Пришвидшений курс Python. Київ: Видавництво Старого Лева, 2021. 600 с.

5. Васильєв О. Програмування мовою Python. Тернопіль: Навчальна книга – Богдан, 2019. 204 с.

6. Ceder N. The Quick Python Book. 3rd ed. New York: Manning Publications, 2018. 432 p.

7. Lambert K. A. Fundamentals of Python: First Programs. Boston: Cengage Learning, 2018. 476 p.

8. Lutz M. Learning Python. 5th ed. Sebastopol: O’Reilly Media, 2019. 648 p.

9. Стрелковська І. В., Григор’єва Т. І., Педяш В. В., Русу О. П., Мельник В. В. Використання Python програмування у графових моделях безпеки баз даних. Наука і техніка сьогодні. 2026. № 1 (55). С. 2570–2580. DOI: 10.52058/2786-6025-2026-1(55)-2570-2580.

Допоміжна

10. Fenner M. Machine Learning with Python for Everyone. Addison-Wesley Data & Analytics Series. Boston: Addison-Wesley Professional, 2019. 586 p.

11. Григор'єва Т., Русу О., Горбачов В., Крохмаль М. Багатовимірний підхід в задачах розробки програмного забезпечення. Вісник Хмельницького національного університету. Серія: Технічні науки. 2025. Т. 359, № 6.2. С. 157–160. DOI: <https://doi.org/10.31891/2307-5732-2025-359-92>

Інформаційні ресурси

12. Бібліотека Міжнародного гуманітарного університету: веб-сайт. URL: <https://mu.od.ua/naukova-biblioteka>

13. Національна бібліотека України ім. В. І. Вернадського: веб-сайт. URL: <http://www.nbuv.gov.ua>