

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ЗАТВЕРДЖУЮ

Декан факультету інформаційних технологій

Тетяна ГОВОРУЩЕНКО

Підпис

Ім'я, ПРІЗВИЩЕ

2025 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Функційне програмування

Назва дисципліни

Призначення Робочої програми

Рівень вищої освіти

Мова навчання

Обсяг дисципліни, кредитів ЄКТС

Статус дисципліни

Факультет

Кафедра

Для освітніх програм різних спеціальностей

Перший (бакалаврський)

Українська

8

Вибіркова

Інформаційних технологій

Комп'ютерної інженерії та інформаційних систем

Форма здобуття освіти	Обсяг дисципліни		Кількість годин						Форма семестрового контролю
			Аудиторні заняття					Самостійна робота (в т.ч. ІРС)	Залік
	Кредити ЄКТС	Години	Разом	Лекції	Лабораторні роботи	Практичні заняття	Семінарські заняття		
Д	8	240	98	32	32	34		142	+

Робоча програма складена на основі освітніх програм підготовки бакалавра та стандарту вищої освіти спеціальності.

Робоча програма складена

Підпис

д-р. філософії Олександр МЕЛЬНИЧЕНКО

Науковий ступінь, вчене звання, Ім'я, ПРІЗВИЩЕ

Схвалена на засіданні кафедри комп'ютерної інженерії та інформаційних систем

Протокол від 29.08 2025 № 1.

Зав. кафедри комп'ютерної інженерії та інформаційних систем

Назва

Підпис

Ольга ПАВЛОВА

Ім'я, ПРІЗВИЩЕ

Хмельницький 2025

3. Пояснювальна записка

Дисципліна «Функційне програмування» належить до циклу вибіркових дисциплін та є важливою для формування у студентів сучасного погляду на розроблення програмного забезпечення. Курс знайомить з парадигмою, що наголошує на використанні чистих функцій та незмінних даних, що дозволяє створювати надійні, масштабовані та легкі для тестування програмні системи. Опанування курсу формує міцну основу для подальшої професійної спеціалізації у таких сферах, як розроблення розподілених систем, аналіз великих даних та штучний інтелект. У процесі навчання здобувачі набувають важливих соціальних навичок (soft skills): здатність до абстрактного мислення, аналізу та синтезу, системного мислення та застосування творчих здібностей для формування нових ідей.

Мета дисципліни. Оволодіння студентами основними концепціями функційного програмування на прикладі мови F#, такими як незмінний стан, функції вищого порядку та чисті функції, а також їх застосуванням для вирішення прикладних задач із залученням методів штучного інтелекту.

Предмет дисципліни. Основи програмування мовою F# з використанням концепцій та парадигм функційного програмування, їх застосування при вирішенні прикладних задач, зокрема у сфері штучного інтелекту.

Завдання дисципліни. Формування практичних навичок програмування мовою F# із застосуванням функційної парадигми; навчання вибору засобів функційного програмування для розроблення прикладних програмних систем; ознайомлення з технікою програмування задач штучного інтелекту з використанням функційного підходу.

Результати навчання. Після вивчення дисципліни студент повинен: *знати* теоретичні положення, що лежать в основі парадигм функційного програмування, основні концепції (незмінний стан, функції вищого порядку, чисті функції, лямбда-обчислення) та принципи програмування задач штучного інтелекту; *володіти* інструментами функційного програмування для реалізації складних алгоритмічних рішень; *вміти* застосовувати набуті знання для проєктування та реалізації програмних систем; *здійснювати* пошук, оброблення та аналіз інформації для розв'язання прикладних задач; *поєднувати* теорію і практику для прийняття оптимальних рішень; *виконувати* експериментальні дослідження та системно мислити для формування нових ідей; *оцінювати* отримані результати та аргументовано захищати прийняті рішення.

4. Структура залікових кредитів дисципліни

Назва розділу (теми)	Кількість годин, відведених на:			
	лекції	лабораторні	практичні	СРС
Тема 1. Основи функційного програмування та F#	8	8	8	34
Тема 2. Робота з колекціями та структурами даних в F#	8	8	8	38
Тема 3. Поглиблені засоби функційного програмування	8	8	8	38
Тема 4. Застосування функційного програмування в машинному навчанні	8	8	8	32
Разом за семестр:	32	32	34	142

5. Програма навчальної дисципліни

5.1 Зміст лекційного курсу

Номер лекції	Перелік тем лекцій, їхні анотації	Кількість годин
	Тема 1. Основи функційного програмування та F#	8
1	Вступ до функційного програмування. Парадигми програмування. Основні принципи ФП: чисті функції, незмінність даних, відсутність побічних ефектів. Переваги функційного підходу. Літ.: [1] с. 9–30; [10] с. 1–20; [12] с. 15–35; [13] с. 25–40.	2
2	Основи мови F#. Синтаксис, типи даних, оператори. Робота з модулями та просторами імен. Виведення типів. Створення першої програми. Літ.: [6] с. 1–40; [10] с. 21–60; [13] с. 41–70.	2
3	Функції у F#. Оголошення функцій. Рекурсія та хвостова рекурсія. Каррінг та часткове застосування функцій. Лямбда-вирази. Літ.: [1] с. 120–156; [6] с. 41–90; [10] с. 61–100; [12] с. 40–70; [13] с. 80–110.	2
4	Системи типів та зіставлення зі зразком. Алгебраїчні типи даних: записи та розмічені об'єднання. Потужність зіставлення зі зразком для оброблення даних. Літ.: [1] с. 300–340; [9] с. 10–24; [10] с. 101–140; [11] с. 5–30; [13] с. 111–150.	2
	Тема 2. Робота з колекціями та структурами даних в F#	8
5	Списки та послідовності. Робота зі списками (List) як основною незмінною колекцією. Лінійні обчислення за допомогою послідовностей (Seq). Літ.: [1] с. 200–230; [6] с. 91–120; [10] с. 141–170; [13] с. 151–180.	2
6	Масиви та інші колекції. Робота зі змінними масивами (Array). Огляд інших колекцій .NET та їх використання у функційному стилі. Літ.: [6] с. 121–150; [10] с. 171–190; [12] с. 180–210; [13] с. 181–200.	2
7	Асоціативні колекції. Робота з незмінними Map та Set. Ефективні операції пошуку, додавання та видалення елементів. Літ.: [1] с. 231–245; [6] с. 151–170; [10] с. 191–205; [13] с. 201–230.	2
8	Проектування функційних структур даних. Реалізація власних незмінних структур даних, таких як стек, черга або дерево. Літ.: [1] с. 246–280; [9] с. 41–70; [12] с. 210–240; [13] с. 231–260.	2
	Тема 3. Поглиблені засоби функційного програмування	8
9	Функції вищого порядку. Використання функцій як аргументів та результатів інших функцій. Функції map, filter, fold для роботи з колекціями. Літ.: [1] с. 156–190; [6] с. 171–200; [10] с. 61–100; [12] с. 241–270.	2
10	Композиція функцій та конвеєризація. Побудова складних операцій за допомогою операторів композиції (>>) та конвеєризації (>). Літ.: Літ.: [1] с. 191–210; [6] с. 200–220; [10] с. 61–80; [12] с. 271–290; [13] с. 90–110.	2
11	Асинхронне та паралельне програмування. Асинхронні робочі процеси (Async). Основи паралельної оброблення даних у функційному стилі. Літ.: Літ.: [5] с. 1–8; [6] с. 240–280; [7] с. 1–7; [12] с. 320–360; [13] с. 300–340.	2
12	Оброблення помилок та тип Option. Функційний підхід до оброблення помилок без винятків. Використання типу Option для представлення відсутності значення. Літ.: Літ.: [1] с. 320–336; [6] с. 221–239; [10] с. 100–120; [11] с. 31–50; [12] с. 290–320; [13] с. 260–290.	2
	Тема 4. Застосування функційного програмування в машинному навчанні	8

Номер лекції	Перелік тем лекцій, їхні анотації	Кількість годин
13	Вступ до машинного навчання з F#. Огляд бібліотек для МН (ML.NET, Accord.NET). Переваги ФП для задач аналізу даних. Літ.: Літ.: [2] с. 1–8; [3] с. 100–112; [4] с. 351–356; [6] с. 280–308; [8] с. 1–10; [12] с. 360–380.	2
14	Підготовка та оброблення даних. Використання функційних підходів для очищення, трансформації та нормалізації даних. Літ.: [3] с. 104–110; [4] с. 355–360; [6] с. 308–320; [8] с. 5–12; [12] с. 380–400.	2
15	Побудова моделей машинного навчання. Реалізація простих моделей (напр., лінійна регресія) у функційному стилі. Робота з API бібліотек. Літ.: Літ.: [3] с. 110–117; [6] с. 320–333; [8] с. 11–18; [12] с. 400–430.	2
16	Оцінка та валідація моделей. Розроблення функцій для оцінки якості моделей. Застосування конвеєрів для створення повного циклу МН. Літ.: Літ.: [2] с. 5–12; [3] с. 112–117; [4] с. 356–360; [6] с. 333–336; [8] с. 18–24; [12] с. 430–448.	2
Разом за семестр:		32

5.2 Зміст практичних занять

№ п/п	Тема практичного заняття	Кількість годин
1	Налаштування середовища розроблення. Перші кроки в F#: F# Interactive, створення проєктів. Літ.: [6] с. 1–20; [10] с. 1–20; [13] с. 25–40.	2
2	Робота з базовими типами та виразами. Розв’язання задач на умовні конструкції та цикли. Літ.: [6] с. 20–50; [10] с. 21–40; [13] с. 41–60.	2
3	Практикум з рекурсії. Розв’язання класичних задач (факторіал, числа Фібоначчі) з хвостовою рекурсією. Літ.: [1] с. 120–156; [9] с. 21–40; [10] с. 61–80; [13] с. 90–110.	2
4	Моделювання доменів за допомогою алгебраїчних типів даних. Літ.: [1] с. 300–330; [6] с. 120–150; [10] с. 101–140; [11] с. 20–50; [13] с. 111–150.	2
5	Маніпуляції зі списками: базові операції, рекурсивне оброблення. Літ.: [1] с. 200–230; [6] с. 90–120; [10] с. 141–170; [13] с. 151–170.	2
6	Ефективна робота з послідовностями для оброблення великих обсягів даних. Літ.: [6] с. 110–140; [10] с. 160–180; [12] с. 240–260; [13] с. 160–180.	2
7	Використання Map та Set для вирішення задач, що потребують унікальності або пошуку за ключем. Літ.: [6] с. 150–170; [10] с. 190–205; [13] с. 200–230.	2
8	Розроблення власної незмінної структури даних (напр., бінарне дерево). Літ.: [1] с. 246–270; [9] с. 41–70; [13] с. 231–260.	2
9	Практичне застосування функцій map, filter, fold для трансформації даних. Літ.: [1] с. 156–190; [10] с. 61–100; [12] с. 241–270.	2
10	Побудова конвеєрів оброблення даних за допомогою оператора > . Літ.: Літ.: [1] с. 191–210; [10] с. 61–80; [12] с. 271–290; [13] с. 90–110.	2
11	Розв’язання задач на асинхронне оброблення даних (напр., завантаження даних з мережі). Літ.: Літ.: [5] с. 1–8; [6] с. 240–280; [7] с. 1–7; [12] с. 320–360; [13] с. 300–340.	2
12	Рефакторинг коду з використанням типу Option для безпечної оброблення відсутніх значень. Літ.: Літ.: [6] с. 221–239; [10] с. 100–120; [11] с. 31–50; [13] с. 260–290.	2
13	Знайомство з датафреймами та бібліотеками для роботи з даними в F#. Літ.: Літ.: [3] с. 100–110; [6] с. 308–320; [8] с. 1–10; [12] с. 380–400.	2
14	Виконання завдань з попереднього оброблення реального набору даних. Літ.: Літ.: [6] с. 308–320; [8] с. 5–12; [12] с. 380–400.	2
15	Тренування простої моделі класифікації або регресії за допомогою ML.NET. Літ.: Літ.: [3] с. 110–117; [6] с. 320–333; [8] с. 11–18; [12] с. 400–430.	2
16	Написання функцій для розрахунку метрик якості моделі (точність, повнота). Літ.: Літ.: [3] с. 112–117; [8] с. 18–24; [12] с. 430–448.	2
Разом за семестр:		34

5.3 Зміст лабораторних занять

№ п/п	Тема лабораторного заняття	Кількість годин
1	Реалізація набору функцій для математичних обчислень з використанням рекурсії та зіставлення зі зразком. Літ.: [1] с. 120–156; [9] с. 21–40; [10] с. 61–80; [13] с. 90–110.	4
2	Створення програми для аналізу текстових даних (підрахунок частоти слів, пошук патернів) за допомогою функцій вищого порядку. Літ.: [1] с. 156–190; [6] с. 171–200; [10] с. 61–100; [12] с. 241–270.	4
3	Розроблення модуля, що реалізує незмінне бінарне дерево пошуку з операціями додавання, пошуку та видалення. Літ.: [1] с. 246–270; [9] с. 41–70; [13] с. 231–260.	4
4	Створення бібліотеки функцій для оброблення даних з активним використанням композиції, каррінгу та конвеєризації. Літ.: [1] с. 191–210; [10] с. 61–80; [12] с. 271–290; [13] с. 90–110.	4
5	Створення програми, яка паралельно обробляє колекцію файлів або асинхронно завантажує дані з кількох веб-ресурсів. Літ.: [5] с. 1–8; [6] с. 240–280; [7] с. 1–7; [12] с. 320–360; [13] с. 300–340.	4
6	Розроблення конвеєру для підготовки табличних даних до машинного навчання: завантаження, очищення, трансформація ознак. Літ.: [3] с. 100–110; [4] с. 355–360; [6] с. 308–320; [8] с. 5–12; [12] с. 380–400.	4
7	Реалізація простого алгоритму машинного навчання. Літ.: [6] с. 320–333; [8] с. 11–18; [12] с. 400–430.	4
8	Розроблення застосунку, що тренує модель на основі бібліотеки ML.NET, оцінює її якість та зберігає для подальшого використання. Літ.: [2] с. 5–12; [3] с. 112–117; [4] с. 356–360; [6] с. 333–336; [8] с. 18–24; [12] с. 430–448.	4
Разом за семестр:		32

5.4 Зміст самостійної роботи здобувача вищої освіти

Самостійна робота студентів полягає у систематичному опрацюванні програмного матеріалу з відповідних джерел інформації, підготовці до практичних та лабораторних занять, виконанні індивідуальних завдань (ІЗ), підготовці до підсумкової контрольної роботи (ПКР). До послуг студентів сторінка кафедри у Модульному середовищі для навчання, де розміщені Робоча програма дисципліни та необхідні матеріали з її навчально-методичного забезпечення та контролю результатів навчання.

Номер тижня	Вид самостійної роботи	Кількість годин
1–2	Опрацювання лекційного матеріалу. Підготовка до виконання та захисту ЛР №1 та підготовка до ПЗ №1–2.	14
3–4	Опрацювання лекційного матеріалу. Підготовка до виконання та захисту ЛР №2 та підготовка до ПЗ №3–4. Виконання ІЗ №1 “Реалізація парсера для простої мови розмітки”.	20
5–6	Опрацювання лекційного матеріалу. Підготовка до виконання та захисту ЛР №3 та підготовка до ПЗ №5–6.	12
7–8	Опрацювання лекційного матеріалу. Підготовка до виконання та захисту ЛР №4 та підготовка до ПЗ №7–8. Виконання ІЗ №2 “Створення функційної бібліотеки для роботи з геометричними фігурами”.	20
9–10	Опрацювання лекційного матеріалу. Підготовка до виконання та захисту ЛР №5 та підготовка до ПЗ №9–10.	12
11–12	Опрацювання лекційного матеріалу. Підготовка до виконання та захисту ЛР №6 та підготовка до ПЗ №11–12. Виконання ІЗ №3 “Розроблення паралельного веб-сканера для збору інформації”.	20
13–14	Опрацювання лекційного матеріалу. Підготовка до виконання та захисту ЛР №7 та підготовка до ПЗ №13–14. Підготовка до ПКР.	20
15–16	Опрацювання лекційного матеріалу. Підготовка до виконання та захисту ЛР №8 та підготовка до ПЗ №15–16. Виконання ІЗ №4 “Побудова та аналіз моделі для прогнозування на основі реального набору даних”. Захист ІЗ.	24
Разом за семестр:		142

Примітки: ЛР – лабораторна робота, ПЗ – практичне заняття, ІЗ – індивідуальне завдання.

На самостійне опрацювання здобувачів виносяться ІЗ з відповідних тем, що сформовані в Модульному середовищі для навчання MOODLE. Керівництво самостійною роботою та контроль за виконанням ІЗ здійснюється викладачем, згідно з розкладом консультацій у позаурочний час.

Вимоги до виконання ІЗ викладені в Модульному середовищі для навчання на сторінці навчальної дисципліни.

6. Технології та методи навчання

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних технологій та методів навчання, зокрема: *лекції* (з використанням проблемного й інтерактивного навчання, інформаційно-комунікаційних технологій); *практичні заняття* (з використанням інструктування, розв'язування типових і прикладних задач, аналізу кейсів); *лабораторні заняття* (з фокусом на програмній реалізації функційних концепцій, тестуванні їхньої ефективності); *самостійна робота* (опрацювання теоретичного матеріалу, підготовка до виконання лабораторних робіт, виконання та захист ІЗ), з використанням інформаційно-комп'ютерних технологій та технологій дистанційного навчання.

7. Методи контролю

Поточний контроль здійснюється під час аудиторних практичних та лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочою програмою і графіком освітнього процесу, зокрема, з використанням Модульного середовища для навчання. Водночас використовуються такі методи поточного контролю:

- оцінювання результатів роботи на практичних заняттях (опитування теоретичного матеріалу, розв'язування задач, участь у обговоренні ситуацій);
- оцінювання результатів захисту лабораторних робіт;
- захист та оцінювання результатів виконання ІЗ.

Підсумкова семестрова оцінка виставляється за результатами поточного контролю. Здобувач вищої освіти, який набрав з будь-якого виду навчальної роботи, суму балів нижчу за 60 відсотків від максимального балу, вважається таким, який *має* академічну заборгованість. Ліквідація академічної заборгованості із семестрового контролю здійснюється у період екзаменаційної сесії або за графіком, встановленим деканатом, відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у ХНУ».

8. Політика дисципліни

Політика навчальної дисципліни загалом визначається системою вимог до здобувача вищої освіти, що передбачені чинними положеннями Університету про організацію і навчально-методичне забезпечення освітнього процесу. Зокрема, проходження інструктажу з техніки безпеки; відвідування занять з дисципліни є обов'язковим. За об'єктивних причин (підтверджених документально) теоретичне навчання за погодженням із лектором може відбуватись в онлайн режимі. Успішне опанування дисципліни і формування фахових компетентностей і програмних результатів навчання передбачає необхідність підготовки до практичних та лабораторних занять (вивчення теоретичного матеріалу з теми, попередню підготовку звіту роботи, підготовку до усного опитування для допуску до заняття), активно працювати на занятті, брати участь у дискусіях щодо прийнятих рішень при виконанні здобувачами лабораторних робіт та виконати ІЗ.

Здобувачі вищої освіти мають дотримуватися встановлених термінів виконання всіх видів навчальної роботи, відповідно до робочої програми навчальної дисципліни. Термін захисту лабораторної роботи вважається своєчасним, якщо студент захистив її на наступному після виконання роботи занятті. Пропущене практичне і лабораторне заняття студент зобов'язаний відпрацювати у встановлений викладачем термін, але не пізніше, ніж за два тижні до кінця теоретичних занять у семестрі.

Засвоєння студентом теоретичного матеріалу з дисципліни оцінюється за результатами опитування під час практичних занять та написання ПКР.

Здобувач вищої освіти, виконуючи самостійну роботу або індивідуальну роботу з дисципліни, має дотримуватися політики доброчесності (заборонені списування, плагіат (зокрема, із використанням мобільних девайсів)). У разі виявлення порушення політики академічної доброчесності в будь-яких видах навчальної роботи здобувач вищої освіти отримує незадовільну оцінку і має повторно виконати завдання з відповідної теми (виду роботи), що передбачені робочою програмою. Будь-які форми порушення академічної доброчесності *не допускаються*.

У межах вивчення навчальної дисципліни здобувачам вищої освіти передбачено визнання і зарахування результатів навчання, набутих через неформальну освіту, що розміщені на доступних платформах, які сприяють формуванню компетентностей і поглибленню результатів навчання, визначених Робочою програмою дисципліни, або забезпечують вивчення відповідної теми та/або виду робіт з програми навчальної дисципліни (детальніше у «Положенні про порядок визнання та зарахування результатів навчання здобувачів вищої освіти у ХНУ»). Так, окремі результати вивчення навчальної дисципліни можуть бути зараховані у випадку отримання студентом результатів навчання у неформальній освіті, що підтверджені відповідним документом (сертифікат, свідоцтво, освітня програма тощо):

- Як результат виконання ЛР № 1–2 зараховується відкритий онлайн-курс:

<https://ocw.tudelft.nl/courses/introduction-to-functional-programming/>

– Як результат виконання ЛР № 3–4 зараховується онлайн-курс: <https://fsharp.org/learn/>

– Як результат виконання ЛР № 5–6 зараховується онлайн-курс:

<https://amplifyingfsharp.io/fsharp-essentials/#:~:text=This%20FREE%20online%20course%20is,way%20you%20develop%20software%20forever>

– Як результат виконання ЛР № 7–8 зараховується онлайн-курс: <https://www.mdft.academy/course-machine-learning-with-fsharp-and-mlnet>

9. Оцінювання результатів навчання студентів у семестрі

Оцінювання академічних досягнень здобувача вищої освіти здійснюється, відповідно до «Положення про контроль і оцінювання академічних досягнень здобувача вищої освіти здійснюється, відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти в ХНУ». При поточному оцінюванні виконаної здобувачем роботи з кожної структурної одиниці та отриманих ним результатів викладач виставляє йому певну кількість балів із встановлених Робочою програмою для цього виду роботи. Водночас кожна структурна одиниця навчальної роботи може бути зарахована, якщо здобувач набрав не менше 60 відсотків (мінімальний рівень для позитивної оцінки) від максимально можливої суми балів, призначеної структурній одиниці.

При оцінюванні результатів навчання здобувачів вищої освіти з будь-якого виду навчальної роботи (структурної одиниці) рекомендується використовувати наведені нижче узагальнені критерії:

Таблиця – Критерії оцінювання навчальних досягнень здобувача вищої освіти

Оцінка та рівень досягнення здобувачем запланованих програмних результатів навчання та сформованих компетентностей	Узагальнений зміст критерія оцінювання
Відмінно (високий)	Здобувач вищої освіти глибоко і у повному обсязі опанував зміст навчального матеріалу, легко в ньому орієнтується і міло використовує понятійний апарат; уміє пов'язувати теорію з практикою, вирішувати практичні завдання, впевнено висловлювати і обґрунтовувати свої судження. Відмінна оцінка передбачає логічний виклад відповіді мовою викладання (в усній або у письмовій формі), демонструє якісне оформлення роботи і володіння спеціальними приладами та інструментами, прикладними програмами. Здобувач не вагається при видозміні запитання, вміє робити детальні та узагальнюючі висновки, демонструє практичні навички з вирішення фахових завдань. При відповіді допустив дві–три несуттєві <i>помилки</i> .
Добре (середній)	Здобувач вищої освіти виявив повне засвоєння навчального матеріалу, володіє понятійним апаратом, орієнтується у вивченому матеріалі; свідомо використовує теоретичні знання для вирішення практичних задач; виклад відповіді грамотний, але у змісті і формі відповіді можуть мати місце окремі неточності, нечіткі формулювання правил, закономірностей тощо. Відповідь здобувача вищої освіти будується на основі самостійного мислення. Здобувач вищої освіти у відповіді допустив дві–три <i>несуттєві помилки</i> .
Задовільно (достатній)	Здобувач вищої освіти виявив знання основного програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, справляється з виконанням практичних завдань, передбачених програмою. Як правило, відповідь здобувача вищої освіти будується на рівні репродуктивного мислення, здобувач вищої освіти має слабкі знання структури навчальної дисципліни, допускає неточності і <i>суттєві помилки</i> у відповіді, вагається при відповіді на видозмінене запитання. Разом з тим, набув навичок, необхідних для виконання нескладних практичних завдань, які відповідають мінімальним критеріям оцінювання і володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді.
Незадовільно (недостатній)	Здобувач вищої освіти виявив розрізнені, безсистемні знання, не вміє виділяти головне і другорядне, допускається помилок у визначенні понять, перекручує їхній зміст, хаотично і невпевнено викладає матеріал, не може використовувати знання при вирішенні практичних завдань. Як правило, оцінка «незадовільно» виставляється здобувачеві вищої освіти, який не може продовжити навчання без додаткової роботи з вивчення навчальної дисципліни.

Структурування дисципліни за видами навчальної роботи і оцінювання результатів навчання студентів денної форми здобуття освіти у семестрі

Аудиторна робота											Контрольні заходи	Самостій на робота	Семестровий контроль
Лабораторні роботи №:								Практичні заняття			Підсумкова контрольна робота (ПКР)	Індивідуа льне завдання (ІЗ)	Залік
1*	2	3	4	5	6	7	8	1	2	3			
Кількість балів за кожний вид навчальної роботи (мінімум-максимум)													
3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5	6-10	6-10	6-10	6-10	12-20	За рейтингом
24-40								18-30			6-10	12-20	60-100

Примітки: *за набрану з будь-якого виду навчальної роботи з дисципліни кількість балів, нижче встановленого мінімуму, здобувач отримує незадовільну оцінку і має її перездати у встановлений викладачем (деканом) термін. Інституційна оцінка встановлюється, відповідно до таблиці «Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС».

Оцінювання на практичних заняттях

Оцінка за активність на практичних заняттях є накопичувальною і складається з таких елементів: усне опитування студентів на знання теоретичного матеріалу з теми; вільне володіння студентом термінологією дисципліни (наприклад, чисті функції, незмінність даних, каррінг, функції вищого порядку) й уміння професійно обґрунтувати прийняті рішення при розв'язуванні задач у функційному стилі; результати виконання завдань під час заняття.

При оцінюванні практичного заняття викладач керується узагальненими критеріями, наведеними у таблиці «**Критерії оцінювання навчальних досягнень здобувача вищої освіти**» (мінімальний позитивний бал – 6 балів, максимальний – 10 балів).

Оцінювання результатів захисту лабораторної роботи

Лабораторна робота, що виконана й оформлена студентом, відповідно до вимог, які визначені в Модульному середовищі для навчання MOODLE, комплексно оцінюється викладачем під час її захисту з урахуванням таких критеріїв: самостійність та коректність програмної реалізації функційних алгоритмів та структур даних; повнота відповіді та знання теоретичних основ (наприклад, принципів ФП, рекурсії, зіставлення зі зразком, роботи з незмінними колекціями); якість програмного коду, його читабельність, дотримання декларативного стилю та обґрунтованість обраних підходів.

Результат виконання і захисту здобувачем вищої освіти кожної лабораторної роботи оцінюється, відповідно до таблиці «**Критерії оцінювання навчальних досягнень здобувача вищої освіти**» (мінімальний позитивний бал – 3 бали, максимальний – 5 балів).

У випадку виявлення здобувачем рівня знань, нижчого ніж 60 відсотків від максимального балу, встановленого Робочою програмою для кожної структурної одиниці, лабораторна робота йому не зараховується і для її захисту він має детальніше опрацювати матеріал з теми роботи, методику її виконання, виправити грубі помилки та повторно вийти на її захист у призначений для цього викладачем час.

Оцінювання підсумкової контрольної роботи

ПКР передбачає виконання трьох завдань (два теоретичних питання та одне практичне завдання, що передбачає розв'язання задачі з розроблення програми у функційному стилі для оброблення набору даних). Під час оцінювання ПКР враховуються: повнота та коректність відповіді, якість вирішення практичного завдання. Кожне теоретичне завдання оцінюється у 2 бали, практичне завдання – у 6 балів; загальна сума балів на позитивну оцінку становить від 6 до 10 балів.

Кількість правильних відповідей	1	2	3
Відсоток правильних відповідей	0-59	60	100
Кількість отриманих балів	3	6	10

У випадку отримання студентом негативної оцінки, ПКР варто перездати до завершення залікового тижня.

Оцінювання результатів виконання індивідуального завдання

ІЗ виконується студентом та оформлюється, відповідно до вимог, які визначені в Модульному середовищі для навчання MOODLE, а також комплексно оцінюється викладачем з урахуванням таких критеріїв: самостійність виконання; правильність розв'язання поставлених задач; обґрунтованість вибору методів розв'язання (наприклад, вибір між рекурсією та функціями вищого порядку, обґрунтування алгебраїчної структури даних); повнота пояснень та аргументованість відповідей; якість оформлення звіту та програмного коду.

Результат виконання здобувачем вищої освіти кожного ІЗ оцінюється, відповідно до таблиці «**Критерії оцінювання навчальних досягнень здобувача вищої освіти**» (мінімальний позитивний бал – 12 балів, максимальний – 20 балів). За підсумками захисту присвоюється відповідна сума балів.

У разі, якщо здобувач вищої освіти виявив рівень знань і виконання ІЗ, що нижчий ніж 60 відсотків від максимальної кількості балів, встановленої Робочою програмою для цієї структурної одиниці, завдання не зараховується. У такому випадку студент має повторно опрацювати зміст завдання, усунути помилки та здати на перевірку доопрацьоване ІЗ у терміни, погоджені з викладачем.

Підсумкова семестрова оцінка

Підсумкова семестрова оцінка за інституційною шкалою і шкалою ЄКТС визначається в автоматизованому режимі після внесення викладачем результатів оцінювання у балах з усіх видів навчальної роботи до електронного журналу. Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС наведені нижче у таблиці «**Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС**».

Таблиця – Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ЄКТС	Рейтингова шкала балів	Інституційна оцінка (рівень досягнення здобувачем вищої освіти запланованих результатів навчання з навчальної дисципліни)	
		Залік	Іспит/диференційований залік
A	90-100	Зараховано	<i>Відмінно/Excellent</i> – високий рівень досягнення запланованих результатів навчання з навчальної дисципліни, що свідчить про безумовну готовність здобувача до подальшого навчання та/або професійної діяльності за фахом
B	83-89		<i>Добре/Good</i> – середній (максимально достатній) рівень досягнення запланованих результатів навчання з навчальної дисципліни та готовності до подальшого навчання та/або професійної діяльності за фахом
C	73-82		
D	66-72		<i>Задовільно/Satisfactory</i> – Наявні мінімально достатні для подальшого навчання та/або професійної діяльності за фахом результати навчання з навчальної дисципліни
E	60-65		
FX	40-59	Незараховано	<i>Незадовільно/Fail</i> – Низка запланованих результатів навчання з навчальної дисципліни відсутня. Рівень набутих результатів навчання є недостатнім для подальшого навчання та/або професійної діяльності за фахом
F	0-39		<i>Незадовільно/Fail</i> – Результати навчання відсутні

Семестровий залік виставляється на останньому занятті за умови, якщо загальна сума балів, яку накопичив здобувач з дисципліни за результатами поточного контролю, перебуває у межах від 60 до 100 балів. Водночас за інституційною шкалою ставиться оцінка «зараховано», а за шкалою ЄКТС – буквене позначення оцінки, що відповідає набраній студентом сумі балів, відповідно до таблиці «**Співвідношення**». Присутність здобувача у цьому випадку не є обов'язковою.

10. Питання для самоконтролю результатів навчання

1. Основні принципи функційної парадигми програмування.
2. Поняття "чистої функції" та її ключові характеристики.
3. Концепція незмінності (immutability) даних та її вплив на розроблення.
4. Відмінності між декларативним та імперативним стилями програмування.
5. Базовий синтаксис мови F# та роль значущих відступів.
6. Механізм виведення типів (type inference) у F#.
7. Поняття рекурсії та її фундаментальна роль у функційному програмуванні.
8. Принцип роботи та переваги хвостової рекурсії для оптимізації.
9. Поняття каррінгу (currying) та його практичне застосування.
10. Часткове застосування функцій (partial application) як інструмент створення нових функцій.
11. Лямбда-вирази (анонімні функції) та сфера їхнього застосування.
12. Алгебраїчні типи даних: записи (records) та їхні властивості.
13. Алгебраїчні типи даних: розмічені об'єднання (discriminated unions) для моделювання доменів.
14. Механізм зіставлення зі зразком (pattern matching) та його можливості.
15. Структура даних List в F#, її незмінність та основні операції.
16. Поняття "лінивих обчислень" (lazy evaluation).
17. Структура даних Seq (последовательность) та її відмінності від списків.
18. Робота з масивами (Array) в F# та сценарії їхнього використання.
19. Використання стандартних колекцій .NET у функційному стилі.

20. Незмінна асоціативна структура даних Map та її основні операції.
21. Незмінна структура даних Set для роботи з унікальними елементами.
22. Порівняльний аналіз ефективності List, Array та Seq для різних задач.
23. Принципи проектування власних незмінних структур даних.
24. Реалізація функційного стека на основі списку.
25. Реалізація функційної черги з використанням списків.
26. Поняття функції вищого порядку.
27. Призначення та використання функції map для трансформації колекцій.
28. Призначення та використання функції filter для фільтрації елементів.
29. Призначення та використання функції fold (згортка) для агрегації даних.
30. Відмінності між лівосторонньою (fold) та правосторонньою (foldBack) згортками.
31. Поняття композиції функцій та використання оператора >>.
32. Принцип роботи конвеєрного оператора |> для побудови послідовностей оброблення.
33. Переваги використання конвеєризації для покращення читабельності коду.
34. Асинхронні робочі процеси (Async) в F# для неблокуючих операцій.
35. Основи паралельного оброблення колекцій для прискорення обчислень.
36. Функційний підхід до оброблення помилок без використання винятків.
37. Тип Option для моделювання наявності або відсутності значення.
38. Тип Result для оброблення операцій, що можуть завершитися успішно або з помилкою.
39. Порівняння підходів до оброблення помилок: винятки, Option та Result.
40. Переваги функційного програмування для задач аналізу даних та машинного навчання.
41. Огляд основних бібліотек для машинного навчання, сумісних з F# (напр., ML.NET).
42. Етапи типового конвеєру (pipeline) машинного навчання.
43. Застосування функційних конвеєрів для підготовки та очищення даних.
44. Поняття ознаки (feature) та цільової змінної (target) у контексті МН.
45. Принцип роботи та реалізація моделі лінійної регресії.
46. Принцип роботи алгоритму k-найближчих сусідів (k-NN) та його реалізація у функційному стилі.
47. Застосування функцій вищого порядку для крос-валідації моделі.
48. Метрики оцінки якості для задач класифікації (точність, повнота, F1-score).
49. Метрики оцінки якості для задач регресії (MAE, MSE, RMSE).
50. Процес збереження та завантаження навчених моделей для їхнього подальшого використання

11. Навчально-методичне забезпечення

Освітній процес з дисципліни «Функційне програмування» повністю і в достатній кількості забезпечений необхідною навчально-методичною літературою, перелік якої наведено у розділі 13.

12. Матеріально-технічне та програмне забезпечення дисципліни

Для проведення лекційних, практичних та лабораторних занять використовуються спеціалізовані комп'ютерні класи кафедри, оснащені сучасною комп'ютерною технікою та мультимедійним обладнанням. Програмне забезпечення включає операційні системи (Windows, Linux), інтегровані середовища розроблення з підтримкою функційного програмування (Visual Studio, Visual Studio Code з плагіном Ionide, JetBrains Rider), .NET SDK з компілятором F# та інструментами для інтерактивного розроблення (F# Interactive), а також системи контролю версій.

13. Рекомендована література

1. Плахта М. Грокаємо функційне програмування / пер. з англ. Ю. Зубкова, В. Зубков. Київ : ArtHuss, 2025. 672 с. URL: <https://www.arthuss.com.ua/shop/hrokayemo-funktsiynе-prohramuvannya> (дата звернення: 28.08.2025).
2. Criticality assessment of wind turbine defects via multispectral UAV fusion and fuzzy logic / P. Radiuk, B. Rusyn, O. Melnychenko, T. Perzynski, A. Sachenko, S. Svystun, O. Savenko. *Energies*. 2025. Vol. 18, no. 17. P. 4523. URL: <https://doi.org/10.3390/en18174523> (date of access: 26.08.2025).
3. Enhanced fire hazard detection in solar power plants: an integrated UAV, AI, and SCADA-based approach / A. Lysyi, A. Sachenko, P. Radiuk, M. Lysyi, O. Melnychenko, O. Ishchuk, O. Savenko. *Radioelectronic and computer systems*. 2025. Vol. 2025, no. 2. P. 99–117. URL: <https://doi.org/10.32620/reks.2025.2.06> (date of access: 28.07.2025).
4. Determining the criticality assessment of defects on wind turbine components detected by UAV sensors / S. Svystun, O. Melnychenko, P. Radiuk, A. Lysyi, A. Sachenko. *Proceedings of the 6th International Workshop on Intelligent Information Technologies & Systems of Information Security (IntelITSIS 2025)* : CEUR-Workshop Proceedings, Khmelnytskyi, Ukraine, 4 April 2025 / ed. by T. Hovorushchenko et al. Aachen, 2025. P. 351–362. URL: <https://ceur-ws.org/Vol-3963/paper28.pdf> (date of access: 28.07.2025).

5. DyTAM: Accelerating wind turbine inspections with dynamic UAV trajectory adaptation / S. Svystun, L. Scislo, M. Pawlik, O. Melnychenko, P. Radiuk, O. Savenko, A. Sachenko. *Energies*. 2025. Vol. 18, no. 7. P. 1823. URL: <https://doi.org/10.3390/en18071823> (date of access: 28.07.2025).
6. Abraham I. *F# in action*. Shelter Island, NY, USA : Manning Publications, 2024. 336 p. URL: <https://www.oreilly.com/library/view/f-in-action/9781633439535/> (date of access: 28.08.2025).
7. Dynamic trajectory adaptation for efficient UAV inspections of wind energy units / S. Svystun, O. Melnychenko, P. Radiuk, O. Savenko, A. Sachenko, A. Lysyi. *Proceedings of the 2024 14th International Conference on Dependable Systems, Services and Technologies (DESSERT)* : Proceedings, Athens, Greece, 11–13 October 2024 / ed. by V. Kharchenko et al. New York, NY, USA, 2025. P. 1–7. URL: <https://doi.org/10.1109/DESSERT65323.2024.11122243> (date of access: 21.08.2025).
8. Intelligent integrated system for fruit detection using multi-UAV imaging and deep learning / O. Melnychenko, L. Scislo, O. Savenko, A. Sachenko, P. Radiuk. *Sensors*. 2024. Vol. 24, no. 6. P. 1913. URL: <https://doi.org/10.3390/s24061913> (date of access: 22.05.2025).
9. Борисенко П. Б., Громова В. В. Програмування мовою Haskell : навч. посіб. до виконання лаборатор. робіт з дисципліни "Функц. програмування". Київ : КПІ ім. Ігоря Сікорського, 2023. 91 с. URL: <https://ela.kpi.ua/server/api/core/bitstreams/4ce1c185-2539-4132-b33f-4147ed8a58b6/content> (дата звернення: 28.08.2025).
10. Russell I. *Essential F#*. Victoria, British Columbia : Lean Publishing, 2023. 207 p. URL: <https://leanpub.com/essential-fsharp> (date of access: 28.08.2025).
11. Статівка Ю. І. Мова програмування Haskell. Базові типи і класи : навч. посібник. Київ : КПІ ім. Ігоря Сікорського, 2023. 78 с. URL: <https://ela.kpi.ua/bitstreams/3030afd8-9244-4fab-b72f-aca5f108c5fe/download> (дата звернення: 28.08.2025).
12. Buonanno E. *Functional programming in C#*. 2nd ed. Shelter Island, NY, USA : Manning Publications, 2022. 448 p. URL: <https://www.oreilly.com/library/view/functional-programming-in/9781617299827/> (date of access: 28.08.2025).
13. Eason K. *Stylish F# 6: crafting elegant functional code for .NET 6*. 2nd ed. Berkeley, CA, USA : Apress, 2021. 422 p. URL: <https://doi.org/10.1007/978-1-4842-7205-3> (date of access: 28.08.2025).

14. Інформаційні ресурси

1. Модульне середовище для навчання. URL: <https://msn.khmnmu.edu.ua/enrol/index.php?id=546>
2. Електронна бібліотека ХНУ. URL: <http://library.khmnmu.edu.ua/>
3. Інституційний репозитарій ХНУ. URL : <https://elar.khmnmu.edu.ua/home>

ФУНКЦІЙНЕ ПРОГРАМУВАННЯ

Тип дисципліни	Вибіркова
Рівень вищої освіти	Перший (бакалаврський)
Мова викладання	Українська
Семестр	—
Кількість призначених кредитів ЄКТС	8,0
Форми здобуття освіти, для яких викладається дисципліна	Очна (денна)

Результати навчання. Після вивчення дисципліни студент повинен: *знати* теоретичні положення, що лежать в основі парадигм функційного програмування, основні концепції (незмінний стан, функції вищого порядку, чисті функції, лямбда-обчислення) та принципи програмування задач штучного інтелекту; *володіти* інструментами функційного програмування для реалізації складних алгоритмічних рішень; *вміти* застосовувати набуті знання для проєктування та реалізації програмних систем; *здійснювати* пошук, оброблення та аналіз інформації для розв'язання прикладних задач; *поєднувати* теорію і практику для прийняття оптимальних рішень; *виконувати* експериментальні дослідження та системно мислити для формування нових ідей; *оцінювати* отримані результати та аргументовано захищати прийняті рішення.

Зміст навчальної дисципліни. Вивчення фундаментальних принципів та практичних підходів до розроблення програмного забезпечення у функційному стилі на прикладі мови F#. Основні теми включають: ключові принципи ФП (чисті функції, незмінність), основи F# (система типів, виведення типів), рекурсію та хвостову рекурсію. Розглядається робота з незмінними колекціями (List, Seq, Map) за допомогою функцій вищого порядку (map, filter, fold). Особлива увага приділяється потужним механізмам, таким як зіставлення зі зразком, композиція функцій та побудова конвеєрів оброблення даних. Курс охоплює поглиблені теми: асинхронне програмування, функційні підходи до оброблення помилок (Option, Result) та застосування функційної парадигми для вирішення задач машинного навчання.

Запланована навчальна діяльність. Мінімальний обсяг навчальних занять в одному кредиті ЄКТС навчальної дисципліни для *першого* (бакалаврського) рівня вищої освіти за денною формою здобуття освіти становить 10 годин на 1 кредит ЄКТС.

Форми (методи) навчання: *лекції* (з використанням проблемного й інтерактивного навчання, інформаційно-комунікаційних технологій); *практичні заняття* (з використанням інструктування, розв'язування типових і прикладних задач, аналізу кейсів); *лабораторні заняття* (з фокусом на програмній реалізації функційних концепцій, тестуванні їхньої ефективності); *самостійна робота* (опрацювання теоретичного матеріалу, підготовка до виконання лабораторних робіт, виконання та захист ІЗ), з використанням інформаційно-комп'ютерних технологій та технологій дистанційного навчання.

Форми оцінювання результатів навчання: оцінювання виконання практичних завдань; оцінювання виконання та захисту лабораторних робіт; оцінювання підсумкової контрольної роботи; оцінювання виконання та захисту індивідуального завдання.

Вид семестрового контролю: залік.

Навчальні ресурси:

1. Плахта М. Грокаємо функційне програмування / пер. з англ. Ю. Зубкова, В. Зубков. Київ : ArtHuss, 2025. 672 с. URL: <https://www.arthuss.com.ua/shop/hrokayemo-funktsiynе-prohramuvannya> (дата звернення: 28.08.2025).
2. Abraham I. F# in action. Shelter Island, NY, USA : Manning Publications, 2024. 336 p. URL: <https://www.oreilly.com/library/view/f-in-action/9781633439535/> (date of access: 28.08.2025).
3. Russell I. Essential F#. Victoria, British Columbia : Lean Publishing, 2023. 207 p. URL: <https://leanpub.com/essential-fsharp> (date of access: 28.08.2025).
4. Стативка Ю. І. Мова програмування Haskell. Базові типи і класи : навч. посібник. Київ : КПІ ім. Ігоря Сікорського, 2023. 78 с. URL: <https://ela.kpi.ua/bitstreams/3030afd8-9244-4fab-b72f-aca5f108c5fe/download> (дата звернення: 28.08.2025).
5. Модульне середовище для навчання. URL: <https://msn.khmnu.edu.ua/enrol/index.php?id=546>
6. Електронна бібліотека ХНУ. Доступ до ресурсу: <http://library.khmnu.edu.ua/>

Викладачі: д-р. філософії, ст. викл. Мельниченко О.В.