

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ЗАТВЕРДЖУЮ
Декан факультету інформаційних технологій
Тетяна ГОВОРУЩЕНКО
09.09.2025

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Мультипарадигмове програмування

Галузь знань F Інформаційні технології
Спеціальність F2 «Інженерія програмного забезпечення»
Рівень вищої освіти Другий (магістерський)
Освітньо-професійна програма Інженерія програмного забезпечення
Обсяг дисципліни 4 кредити ЄКТС
Шифр дисципліни ОФП.01
Мова викладання українська
Статус дисципліни: обов'язкова (фахової підготовки)

Форма здобуття освіти	Курс	Семестр	Обсяг дисципліни		Кількість годин						Форма семестрового контролю
			Кредити ЄКТС	Години	Аудиторні заняття					Самостійна робота (в т.ч. ІРС)	Іспит
					Разом	Лекції	Лабораторні роботи	Практичні заняття	Семінарські заняття		
Д	1	2	4	120	34	16	18			86	+

Робоча програма складена на основі освітньо-професійної програми «Інженерія програмного забезпечення» за спеціальністю F2 «Інженерія програмного забезпечення»

Робоча програма складена Підпис(и) автора(ів) канд.техн.наук, доцент, Оксана ЯШИНА
Науковий ступінь, учене звання, Ім'я, ПРІЗВИЩЕ автора(ів)

Схвалена на засіданні кафедри Інженерії програмного забезпечення

Протокол від 28 серпня 2025 №1.

Зав. кафедри Підпис Леонід БЕДРАТЮК
Ім'я, ПРІЗВИЩЕ

Робоча програма розглянута та схвалена вченою радою факультету інформаційних

Голова вченої ради факультету Підпис Тетяна ГОВОРУЩЕНКО
Ім'я, ПРІЗВИЩЕ

Хмельницький 2025

2 ЛИСТ ПОГОДЖЕННЯ

Посада	Назва факультету	Підпис	Ініціали, прізвище
Завідувач кафедри інженерії програмного забезпечення, доктор, фіз.-мат. наук, проф.	Факультет інформаційних технологій		Леонід БЕДРАТЮК
Гарант освітньо-професійної програми, к.т.н., доцент	Факультет інформаційних технологій		Оксана ЯШИНА

МУЛЬТИПАРАДИГМОВЕ ПРОГРАМУВАННЯ

Тип дисципліни	Обов'язкова
Рівень вищої освіти	Другий (магістерський)
Мова викладання	Українська
Семестр	Другий
Кількість встановлених кредитів ЄКТС	4,0
Форми здобуття освіти, для яких викладається дисципліна	Очна (денна)

Результати навчання.

Студент, який успішно завершив вивчення дисципліни, повинен: обґрунтовано *обирати* парадигми і мови програмування, в тому числі мультипарадигмові, для вирішення прикладних завдань; *застосовувати* на практиці системні та спеціалізовані засоби, компонентні технології (платформи) та інтегровані середовища розробки програмного забезпечення; *аналізувати, оцінювати, розробляти і обирати* методи, сучасні програмно-апаратні інструментальні та обчислювальні засоби, технології, алгоритмічні та програмні рішення для ефективного виконання конкретних виробничих задач з програмної інженерії; *аналізувати, оцінювати і обирати* методи, сучасні програмно-апаратні інструментальні та обчислювальні засоби, технології, алгоритмічні та програмні рішення для ефективного виконання конкретних виробничих задач з програмної інженерії.

Зміст навчальної дисципліни. Поняття про мультипарадигмове програмування. Методика порівняння мов програмування. Використання мультипарадигмового підходу. Використання мультипарадигмового підходу. Мультипарадигмова розробка та мова F#. Засоби функціонального програмування. Прикладне функціональне програмування. Створення нової мови програмування. Розширення існуючої мови програмування. Інтерпретатори: розширені та вбудовані. Трансляція з однієї мови на іншу. Конструювання модулів, написаних на різних мовах програмування. Розширення мови програмування. Бібліотечне розширення існуючої мови програмування.

Пререквізити – ОЗП.03 Англійська мова за професійним спрямуванням, ОФП.02 Методології та технології розробки програмних систем, ОФП.06 Реінжиніринг та оптимізація програмних систем.

Кореквізити – ОФП.07 Переддипломна практика, ОФП.08 Кваліфікаційна робота.

Запланована навчальна діяльність: мінімальний обсяг навчальних занять в одному кредиті ЄКТС навчальної дисципліни для другого (магістерського) рівнів вищої освіти за денною формою здобуття освіти становить 8 годин.

Форми (методи) навчання: лекції (з використанням методів проблемного навчання і візуалізації), лабораторні заняття (з використанням інтерактивних, інформаційно-комунікаційних технологій), практичні заняття (з використанням частково-пошукових вправ, кейсів), самостійна робота (опрацювання лекційного матеріалу, проходження тестування).

Форми оцінювання результатів навчання: захист лабораторних робіт, тестування.

Вид семестрового контролю: іспит – 2 семестр.

Навчальні ресурси:

1. Petra Hofstedt. Realizing Evaluation Strategies by Hierarchical Graph Rewriting. In Germán Vidal and Neng-Fa Zhou, editors, Implementation of Constraint Logic Programming Systems – CICLOPS, 2020.
2. Парадигми програмування. URL: <https://w3schoolsua.github.io/hyperskill/paradigms.html#gsc.tab=0>.
3. Peter Van Roy. Діаграма з результатами порівняння більше 30-ти парадигм програмування. <https://www.info.ucl.ac.be/~pvr/paradigmsDIAGRAMeng108.pdf>.
4. Модульне середовище. URL: <https://msn.khmnua.edu.ua/>.
5. Електронна бібліотека. URL: <http://library.khmnua.edu.ua/>.

Викладач: канд. техн. наук, доцент кафедри ІТЗ Яшина О.М.

3. ПОЯСНЮВАЛЬНА ЗАПИСКА

Дисципліна «Управління проектами і якістю програмних систем» є однією із фахових дисциплін і займає провідне місце у підготовці фахівців освітнього рівня «магістр» за спеціальністю F2 «Інженерія програмного забезпечення» за освітньо-професійною програмою «Інженерія програмного забезпечення».

Пререквізити – ОЗП.03 Англійська мова за професійним спрямуванням, ОФП.02 Методології та технології розробки програмних систем, ОФП.06 Реінжиніринг та оптимізація програмних систем.

Кореквізити – ОФП.07 Переддипломна практика, ОФП.08 Кваліфікаційна робота.

Відповідно до освітньої програми дисципліна має забезпечити:

компетентності – ІК. Здатність особи розв'язувати складні задачі і проблеми у певній галузі професійної діяльності або у процесі навчання, що передбачає проведення досліджень та/або здійснення інновацій та характеризується невизначеністю умов і вимог. ЗК.01. Здатність до абстрактного мислення, аналізу та синтезу. ФК.01. Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення. ФК.2. Здатність розробляти і реалізовувати наукові та/або прикладні проекти у сфері інженерії програмного забезпечення. ФК.5. Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення. ФК.07. Здатність критично осмислювати проблеми у галузі інформаційних технологій та на межі галузей знань, інтегрувати відповідні знання та розв'язувати складні задачі у широких або мультидисциплінарних контекстах. ФК.8. Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення. УК.01. Здатність проектувати програмні системи та їх компоненти із використанням поглиблених знань з оптимізації, реінжинірингу програмних систем, управління проектами та якістю з вибором парадигм програмування та застосування машинного навчання.

програмні результати навчання: ПРН.07. Аналізувати, оцінювати і застосовувати на системному рівні сучасні програмні та апаратні платформи для розв'язання складних задач інженерії програмного забезпечення. ПРН.09. Обґрунтовано вибирати парадигми і мови програмування для розроблення програмного забезпечення; застосовувати на практиці сучасні засоби розроблення програмного забезпечення. ПРН.14. Прогнозувати розвиток програмних систем та інформаційних технологій. ПРН.18. Вміти використовувати методи фундаментальних і прикладних дисциплін інженерії програмного забезпечення при проектуванні архітектури та розробці програмних систем із використанням поглиблених знань з оптимізації, реінжинірингу програмних систем, управління проектами та якістю з вибором парадигм програмування та застосуванням машинного навчання.

Мета дисципліни: опанування студентами компонентних технологій та інтегрованих середовищ розробки для ефективного виконання конкретних виробничих задач програмної інженерії із обґрунтованим вибором парадигм програмування та мультипарадигмових мов програмування.

Предмет дисципліни:

Предметом вивчення дисципліни є парадигми програмування та принципи мультипарадигмового програмування.

Завдання дисципліни:

Надати студентам знання і практичні навички практичного вирішення прикладних задач з обґрунтованим вибором мов програмування, в тому числі і мультипарадигмових.

4. СТРУКТУРА ЗАЛІКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ
«Мультипарадигмове програмування»

Назва теми	Лекції	Лаб. роботи	Самостійна робота здобувачів
Тема1. Вступ. Основи мультипарадигмового програмування.	4	8	20
Тема 2. Мультипарадигмове проектування.	4	4	20
Тема 3. Функціональний парадигма програмування.	4	2	20
Тема 4. Створення нової мови програмування.	4	4	26
Разом за 2-й семестр	16	18	86

5. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

«Мультипарадигмове програмування»

5.1. Зміст лекційного курсу

№ лекції	Перелік тем лекцій, їх анотації	К-сть годин
1	2	3
1	Лекція 1. Вступ. Основи мультипарадигмового програмування. Поняття про парадигми програмування. Еволюція парадигм програмування. Сутність, види, класифікація парадигм програмування. Суть мультипарадигмового підходу до програмування. Введення поняття процедур та абстракції даних. Характеристики та концепції первинних парадигм програмування, імперативних та декларативних (включаючи функціональну та реляційну). Процедурне та модульне програмування. Літ.: [2] с.36-78	2
2	Лекція 2. Методика порівняння мультипарадигмових мов програмування Опис сучасних мультипарадигмових мов програмування. Їх класифікація та порівняння. Представлення парадигмальних особливостей мов програмування. Результати парадигмального аналізу. Літ.: [1] с.36-78	2
3	Лекція 3. Інженерія предметної області в контексті парадигм програмування Необхідність кількох парадигм проектування. Визначення мультипарадигмового проектування, аналіз, сімейства, предметні області; прикладний аналіз та аналіз рішень. Сімейства та аналіз спільності. Вимірювання абстракції, точність абстракції. Мультипарадигмове проектування та реалізація. Складні сімейства та включення патернів. Літ.: [1,3]	2
4	Лекція 4. Використання мультипарадигмового підходу в проектуванні програмного забезпечення. Огляд методів мультипарадигмового проектування. Міри складності в проектуванні програмних систем. Проектування з кількома предметними областями в єдиній парадигмі та з різними парадигмами. Кілька парадигм в кожній підобласті. Циклічні підобласті. Операції мультипарадигмового проектування. Переплітання парадигм - аналіз спільності, вимірювання спільності; аналіз мінливості. Взаємозалежні предметні області. Архітектура мультипарадигмового проектування. Рішення які базуються на патернах. Значення ідіом та патернів. Типові патерни - міст, метод шаблонів, стан, уніфікація, синглтон, адаптер. Мультипарадигмове проектування як доповнення патернів.	2
5	Лекція 5. Мультипарадигмова розробка та аналіз прикладної предметної області Мінливість в часі і просторі. Динамічне зв'язування. Дедуктивна та індуктивна спільність. Сімейства програм. Словник предметної області. Аналіз мінливості. Позитивна та негативна мінливість. Предметна область та діапазон мінливості. Час зв'язування та гнучкість. Ефективність та час зв'язування; варіанти зв'язування. Таблиці мінливості. Графи залежності мінливості. Мультипарадигмовий аналіз прикладної предметної області та	2

	області рішень. Сімейства систем, члени сімейства в прикладній області і області рішень. Баланс загального і часткового. Рівні предметної абстракції. Аналіз прикладної області і області рішень. Підобласті в предметному аналізі. Предметний аналіз та багаторазове використання. Модульна організація підобластей. Повторення та ієрархія, структура підобласті; каркас. Літ.: [1]	
6	Лекція 6. Використання засобів функціонального програмування в мультипарадигмовому аспекті. Функції, які повертають функції. Взаємна рекурсія. Символьні оператори. Іменовані зразки. Зіставлення з літералами. Обмеження when. Групування зразків. Зіставлення структур даних. Необов'язкові значення. Зіставлення з груповими символами. Альтернативний синтаксис лямбда-виразів. Розмічені об'єднання. Використання розмічених об'єднань для створення деревовидних структур. Методи і властивості розмічених об'єднань. Записи. Типи Lazy. Літ.: [1,3]	2
7.	Лекція 7. Прикладне функціональне програмування в аспекті мультипарадигменості. Одиниці виміру. Активні шаблони. Використання модулів. Перетворення модулів в класи. Робота зі списками. Хвостова рекурсія. Програмування з застосуванням функцій. Функціональні шаблони проектування. Літ. [2,4,6]	2
8	Лекція 8. Конструювання модулів. Розширення існуючої мови програмування. Вибір складових мови програмування. Інтерпритатор, компілятор, парсер. Особливості роботи із компіляторами та інтерпритаторами. Створення компілятора. Використання методу транспайлінгу. Інтерпретатори: розширені та вбудовані. Трансляція з однієї мови на іншу. Конструювання модулів, написаних на різних мовах програмування. Класифікація розширень мов програмування. Розширення доступні через API. Огляд методів розширення мов програмування. Розширення за допомогою мовних конструкцій. Принципи використання бібліотек для розширення мов програмування. Написання модулів розширення. Літ.: [1]	2
	Разом за 2-й семестр	16

5.2. Зміст лабораторних занять

№ з/п	Тема лабораторного заняття	Кількість годин
1	Лабораторна робота №1. Частина 1 Порівняння мов програмування в аспекті мультипарадигмового програмування Літ.: [1] с.45-132, [2] с.14-140, [3] с.25-92, [4, 5]	2
2	Лабораторна робота №1. Частина 2 Порівняння мов програмування в аспекті мультипарадигмового програмування Літ.: [1] с.45-132, [2] с.14-140, [3] с.25-92, [4, 5]	2
3	Лабораторна робота №2. Частина 1 Здійснення парадигмального аналізу Літ.: [1] с.169-189, [2] с.14-140, [3] с.93-186, [4, 5]	2
4	Лабораторна робота №2. Частина 2 Здійснення парадигмального аналізу Літ.: [1] с.169-189, [2] с.14-140, [3] с.93-186, [4, 5]	2
5	Лабораторна робота №3. Частина 1 Вибір складових мови програмування Літ.: [7] с.305-330	2
6	Лабораторна робота №3. Частина 2 Вибір складових мови програмування Літ.: [7] с.305-330	2
7	Лабораторна робота №4. Частина 1 Розробка інтерпретатора засобами F# Літ.: [7] с.392-418, [8] с.220-253	2
8	Лабораторна робота №4. Частина 2 Розробка інтерпретатора засобами F# Літ.: [7] с.392-418, [8] с.220-253	2
9	Підсумкове заняття	2
	Разом за семестр	18

5.3 Зміст самостійної роботи

Об'єм самостійної роботи з дисципліни “Основи командної розробки програмного забезпечення” становить 86 годин. Вони включають опрацювання лекційного матеріалу, теоретичних і лабораторних завдань, підготовку до виконання лабораторних робіт, їх захисту і поточне тестування.

Самостійна робота здобувачів вищої освіти полягає у систематичному опрацюванні програмного матеріалу з відповідних джерел інформації, підготовці до лабораторних занять, виконанні індивідуальних завдань, тестування з теоретичного матеріалу тощо. Крім цього до послуг студентів сторінка навчальної дисципліни у Модульному середовищі для навчання, де розміщені Робоча програма дисципліни та необхідні документи з її навчально-методичного забезпечення.

Номер тижня	Зміст самостійної роботи	Кількість годин
1	2	3
1	Тема 1. Вступ. Основи мультипарадигмового програмування. Опрацювання лекційного матеріалу, підготовка до виконання лабораторної роботи № 1.	6
2	Тема 1. Вступ. Основи мультипарадигмового програмування. Опрацювання лекційного матеріалу Підготовка до захисту лабораторної роботи №1	6
3	Тема 1. Вступ. Основи мультипарадигмового програмування. Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №2. Опрацювання довідкової літератури.	6
4	Тема 1. Вступ. Основи мультипарадигмового програмування. Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи № 2	2
5	Тема 2. Мультипарадигмове проектування. Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи № 3. Опрацювання довідкової літератури.	6
6	Тема 2. Мультипарадигмове проектування. Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи № 3. Опрацювання довідкової літератури.	6
7	Тема 2. Мультипарадигмове проектування. Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи № 4. Опрацювання довідкової літератури.	6
8	Тема 2. Мультипарадигмове проектування. Опрацювання лекційного матеріалу та довідкової літератури. Підготовка до ТК.	2
9	Тема 3. Функціональний парадигма програмування. Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи № 4. Опрацювання довідкової літератури.	6
10	Тема 3. Функціональний парадигма програмування. Опрацювання лекційного матеріалу. Підготовка до виконання практичної роботи № 1. Опрацювання довідкової літератури.	6
11	Тема 3. Функціональний парадигма програмування. Опрацювання лекційного матеріалу. Підготовка до виконання практичної роботи №2. Опрацювання довідкової літератури.	6
12	Тема 3. Функціональний парадигма програмування. Опрацювання лекційного матеріалу. Підготовка до захисту практичної роботи № 1. Опрацювання довідкової літератури.	2
13	Тема 4. Створення нової мови програмування. Опрацювання лекційного матеріалу. Підготовка до захисту практичної роботи № 2. Опрацювання довідкової літератури.	6
14	Тема 4. Створення нової мови програмування. Опрацювання лекційного матеріалу. Підготовка до виконання практичної роботи № 3. Опрацювання довідкової літератури.	6

15	Тема 4. Створення нової мови програмування. Опрацювання лекційного матеріалу. Підготовка до захисту практичної роботи №3. Опрацювання довідкової літератури.	6
16	Тема 4. Створення нової мови програмування. Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи та захисту лабораторної роботи №4. Опрацювання довідкової літератури.	2
Разом за 2-й семестр		86

На самостійне опрацювання студентів виносяться визначені у методичних рекомендаціях до лабораторних занять та самостійної роботи питання з кожної теми. Керівництво самостійною роботою та контроль за виконанням індивідуального завдання здійснюється викладачем згідно з розкладом консультацій у позаурочний час.

Вимоги до виконання контрольної роботи (для студентів заочної форми здобуття освіти) та індивідуального домашнього завдання (для студентів денної форми здобуття освіти) 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, зокрема: лекції (з використанням методів проблемного навчання і візуалізації); лабораторні заняття (бесіда, демонстрування, спостереження, з використанням кейсів, розв'язування задач, презентацій), самостійна робота (індивідуальне домашнє завдання, опрацювання теоретичного матеріалу).

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних технологій та методів навчання, зокрема: методи навчання за джерелом передачі і сприймання інформації (словесні (пояснення, дискусія, консультування), практичні (інструктування, розв'язування ситуаційних задач), наочні (демонстрування, ілюстрування, спостереження); за логікою передачі і сприймання навчальної інформації; за рівнем самостійності пізнавальної діяльності (методи проблемного викладу, частково пошукові, дослідницькі); методи стимулювання і мотивації учіння, інтерактивні; метод аналізу конкретних ситуацій (case-study) з використанням технологій візуалізації, інформаційно-комунікаційних та технології дистанційного навчання (сервіс для проведення онлайн конференцій Zoom, Модульне середовище для навчання тощо).

7. Методи контролю

Поточний контроль здійснюється під час практичних занять, а також у дні проведення контрольних заходів, встановлених робочою програмою і графіком освітнього процесу.

При цьому використовуються такі методи поточного контролю:

- тестовий контроль теоретичного матеріалу;
- оцінювання результатів роботи на лабораторних заняттях (розв'язування задач, участь у обговоренні ситуацій, самостійні роботи);

При виведенні підсумкової семестрової оцінки враховуються результати як поточного контролю, так і підсумкового контролю, який проводиться з усього матеріалу дисципліни за білетами, попередньо розробленими і затвердженими на засіданні кафедри. Здобувач вищої освіти, який набрав з будь-якого виду навчальної роботи, суму балів нижчу за 60 відсотків від максимального балу, не допускається до семестрового контролю, поки не виконає обсяг роботи, передбачений Робочою програмою. Здобувач вищої освіти, який набрав позитивний середньозважений бал (60 відсотків і більше від максимального балу) з усіх видів поточного контролю і не склав іспит, 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 індивідуальному режимі. Успішне опанування дисципліни і формування фахових компетентностей і програмних результатів навчання передбачає необхідність підготовки до практичних занять (вивчення теоретичного матеріалу з теми, активно працювати на занятті, розв'язувати задачі, брати участь у дискусіях щодо прийнятих рішень при виконанні здобувачами задач).

Здобувачі вищої освіти зобов'язані дотримуватися термінів виконання усіх видів робіт у встановлені терміни, передбачених робочою програмою навчальної дисципліни. Пропущене практичне заняття здобувач 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, тестування й виконання індивідуального домашнього завдання.

Здобувач вищої освіти, виконуючи самостійну роботу з дисципліни, має дотримуватися політики доброчесності (заборонені списування, підказки, плагіат, використання штучного інтелекту (без вірного цитування)). У разі 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 під час вивчення навчальної дисципліни не допускаються та не толеруються.

У межах вивчення навчальної дисципліни здобувачам вищої освіти передбачено визнання і зарахування результатів навчання, набутих шляхом неформальної освіти, що розміщені на доступних платформах (<https://github.com>, <https://docs.oracle.com/javase/tutorial/java/index.html>, <http://www.bruceeckel.com/>, <http://java.sun.com/products/jdk/1.3/docs/index.html>, <http://java.sun.com/products/jdk/1.3/devdocs-vs-specs.html>, <http://developer.java.sun.com/developer/onlineTraining/>, <http://java.sun.com/products/java-media/jmf/2.1.1/guide/JMFTOC.html>), 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).

9. Оцінювання результатів навчання студентів у семестрі

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

Будь-які форми порушення академічної доброчесності **не допускаються та не толеруються**.

Отриманий здобувачем бал за зарахований вид навчальної роботи (структурну одиницю) після її оцінювання викладач виставляє в електронному журналі обліку успішності здобувачів вищої освіти. За умови виконання усіх видів навчальної роботи за результатами поточного контролю протягом вивчення навчальної дисципліни, встановлених її Робочою програмою, здобувач денної форми здобуття освіти з навчальної дисципліни, підсумковим

контролем для якої є іспит, може набрати до 60 балів (здобувач заочної форми – до 50 балів). Позитивну підсумкову оцінку здобувач може отримати, якщо за результатами поточного та підсумкового контролів набере від 60 до 100 балів. Семестрова підсумкова оцінка розраховується в автоматизованому режимі в інформаційній підсистемі «Електронний журнал» (ІС «Електронний університет») і відповідно до накопиченої суми балів визначається оцінка за інституційною шкалою та шкалою ЄКТС (див. таблицю Співвідношення...), яка заноситься в екзаменаційну відомість, а також до Індивідуального навчального плану здобувача вищої освіти.

Таблиця – Критерії оцінювання навчальних досягнень здобувача вищої освіти

Оцінка та рівень досягнення здобувачем запланованих ПРН 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 вміло 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>похибки</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 їх 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	Семестровий контроль	Разом
<i>Другий семестр</i>										
Лабораторні роботи №:								Тестовий контроль:	Іспит	Сума балів
1	2	3	4	5	6	7	8	Т 1		

Кількість балів за вид навчальної роботи (мінімум-максимум)									
3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5	12-20	
24-40								12-20	24-40 60-100

Умовні позначення: Т – тема навчальної дисципліни

Оцінювання результатів захисту лабораторної роботи

Виконана й оформлена відповідно до встановлених Методичними рекомендаціями вимог лабораторна робота комплексно оцінюється викладачем при її захисті з урахуванням таких критеріїв: самостійність та правильність виконання; повнота відповіді та знання методики проектування та розробки коду (застосунок); наявність відповідей на контрольні питання до відповідної лабораторної роботи, наявність звіту зі скріпками коду та результатів його роботи, тощо.

Результат виконання і захисту здобувачем вищої освіти кожної лабораторної роботи оцінюється відповідно до таблиці Критеріїв оцінювання навчальних досягнень здобувача вищої освіти та рівня досягнення здобувачем запланованих ПРН та сформованих компетентностей з присвоєнням йому відповідної суми 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 результатів навчання з теоретичного матеріалу (тестовий контроль)

Тест, передбачений Робочою програмою, складається із 30 тестових завдань, кожне з яких є рівнозначним. Максимальна сума балів, яку може набрати студент, складає 100 (кількість набраних балів за тестове завдання може бути різною).

Відповідно до таблиці структурування видів робіт за тематичний контроль здобувач залежно від кількості правильних відповідей може отримати від 60 до 100 балів.

Розподіл балів в залежності від наданих правильних відповідей на тестові завдання

Кількість правильних відповідей	1-17	18-23	24-26	27-30
Відсоток правильних відповідей	0-59	60-79	80-89	90-100
Кількість балів	-	3	4	5

На тестування відводиться 30 хвилин. Студент проходить тестування в он-лайн режимі у Модульному середовищі для навчання. При отриманні негативної оцінки тест слід перездати до терміну *наступного* контролю.

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	Відмінно/Excellent – 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		Добре/Good – 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		Задовільно/Satisfactory – Наявні мінімально достатні для подальшого навчання та/або професійної діяльності за фахом результати навчання з навчальної дисципліни
D	66-72		
E	60-65		
FX	40-59	Незараховано	Незадовільно/Fail – 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

10. Питання до тестового контролю студентів з дисципліни «Мультипарадигмове програмування»

1. Поняття про простори імен у F#.
2. Управління файлами з вихідними кодами F#.
3. Значення пробільних символів.
4. Взаємодія з бібліотеками .NET.
5. Прив'язка let.
6. Поняття про функції у F#.
7. Параметри функцій у F#. Виведення типів.
8. Узагальнені функції.
9. Область видимості.
10. Що таке каррінг.
11. Рекурсивні функції.
12. Функціональні значення.
13. Лямбда-вирази.
14. Композиція функцій. Прямий оператор композиції. Зворотний оператор композиції.
15. Конвеєризація. Прямий конвеєрний оператор. Зворотний конвеєрний оператор.
16. Типи мови F#. Синтаксис типів. Функції перетворення типів.
17. Арифметика F#. Бітові операції. Символи. Рядки.
18. Порівняння і рівність.
19. Булеві значення.

11. Питання для самоперевірки та іспиту з дисципліни «Мультипарадигмове програмування»

1. Поняття про мультипарадигмові мови програмування.
2. Парадигми програмування.
3. Дві парадигми програмування.
4. Три парадигми програмування.
5. Чотири парадигми програмування.
6. Функціональна парадигма програмування.
7. Об'єктно-орієнтована парадигма програмування.
8. Логічна парадигма програмування.
9. Рефлексивна парадигма програмування.
10. Основи створення нових мов програмування.
11. Вибір інтерпретатора та компілятора.
12. Розробка інтерпретатора та компілятора.
13. Розширення мов програмування.

14. Вказати методи розширення мов програмування.
15. Використання різних методів для розширення мов програмування.
16. Будова програми на мові F#.
17. Дати визначення модуля. Поняття про вкладені модулі.
18. Поняття про простори імен у F#.
19. Управління файлами з вихідними кодами F#.
20. Значення пробільних символів.
21. Взаємодія з бібліотеками .NET.
22. Прив'язка let.
23. Поняття про функції у F#.
24. Параметри функцій у F#. Виведення типів.
25. Узагальнені функції.
26. Область видимості.
27. Що таке каррінг.
28. Рекурсивні функції.
29. Функціональні значення.
30. Лямбда-вирази.
31. Композиція функцій. Прямий оператор композиції. Зворотний оператор композиції.
32. Конвеєризація. Прямий конвеєрний оператор. Зворотний конвеєрний оператор.
33. Типи мови F#. Синтаксис типів. Функції перетворення типів.
34. Арифметика F#. Бітові операції. Символи. Рядки.
35. Порівняння і рівність.
36. Булеві значення.
37. Умовні вирази.
38. Вираз while ... do.
39. Прості цикли for. Вираз for ... to.
40. Цикли-перерахування for. Вираз for ... in.
41. Зіставлення зі зразком.
42. Кортежі.
43. Списки.
44. Діапазони списків.
45. Генератори списків.
46. Функції модуля List.
47. Агрегатні оператори модуля List.
48. Згортки.
49. Тип option.
50. Послідовності та вирази послідовності.
51. Функції модуля Seq.
52. Агрегатні оператори модуля Seq.
53. Масиви. Зрізи масивів.
54. Способи створення масивів.
55. Функції модуля Array.

56. Агрегатні оператори модуля Aggru.
57. Багатовимірні масиви. Прямокутні масиви. Невирівняні масиви.
58. Функції, які повертають функції.
59. Взаємна рекурсія.
60. Символьні оператори.
61. Іменовані зразки.
62. Зіставлення з літералами.
63. Обмеження when.
64. Групування зразків.
65. Зіставлення структур даних.
66. Необов'язкові значення.
67. Зіставлення з груповими символами.
68. Альтернативний синтаксис лямбда-виразів.
69. Розмічені об'єднання.
70. Використання розмічених об'єднань для створення деревовидних структур.
71. Методи і властивості розмічених об'єднань.
72. Записи.
73. Типи Lazy.
74. Методи штучного інтелекту.
75. Машинне навчання.

12. Навчально-методичне забезпечення

Освітній процес з дисципліни «Управління проектами і якістю програмних систем» повністю і в достатній кількості забезпечений необхідною навчально-методичною літературою. Зокрема, викладачами кафедри підготовлені і видані такі роботи:

Мультипарадигмове програмування. Лабораторний практикум для студентів спеціальності “Інженерія програмного забезпечення” / О. М. Яшина. – Хмельницький : ХНУ, 2021. – 65 с.

13. Матеріально-технічне та програмне забезпечення дисципліни (за потреби)

Інформаційна та комп'ютерна підтримка: ПК, планшет, смартфон або інший мобільний пристрій, проєктор. Програмне забезпечення: програми Microsoft Office або аналогічні, доступ до мережі Інтернет, робота з презентаціями.

Вивчення навчальної дисципліни не потребує використання спеціального програмного прикладного забезпечення, крім загальноновживаних програм і операційних систем.

14. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Peter Van Roy. Діграма з результатами порівняння більше 30-ти парадигм програмування. URL: <https://www.info.ucl.ac.be/~pvr/paradigmsDIAGRAMeng108.pdf>
2. Aleksei Sedunov Kotlin In-Depth [Vol-I]: A Comprehensive Guide to Modern Multi-Paradigm Language / USA: BPB Publications, 2019. – 348 p.
3. Bedir Tekinerdogan Multi-Paradigm Modelling Approaches for Cyber-Physical Systems / USA: Academic Press, 2020. – 314 p.
4. Chris Smith Programming F# 3.0 / USA: O'Reilly Media, 2022. – 476 p.

5. Joseph Fernandez Learn F#: Functional Programming / ROI Press, 2020. – 152 p.
6. Wiki Books F# Programming / USA: opensource-books, 2021. – 436 p.
7. Michael Metcalf, John Reid, Reid Metcalf The F Programming Language / USA: Oxford University Press, 2016. – 344 p.
8. Daniel Holden Build Your Own Lisp / USA: ebook publishing, 2024. – 276 p.
9. О. Перевозчикова Інформаційні системи і структури даних. – К. : К-М Академія. – 2011. – 288 с.
10. Д. Сідоров, І. Казак Основи інженерних розрахунків на ПЕОМ. Програмування алгоритмічною мовою Фортран. – В: ЦНЛ. – 186 с.
11. Говорущенко Т. О. Методологія оцінювання достатності інформації для визначення якості програмного забезпечення: монографія. Хмельницький: Хмельницький національний університет, 2017. – 310 с.
12. Бевз О. М. Проєктування програмних засобів систем управління: навчальний посібник / О. М. Бевз, В. М. Папінов, Ю. А. Скидан. – Вінниця: ВНТУ, 2020. – 125 с.
13. Говорущенко Т. О. Аналіз, дослідження та оцінювання програмних систем: навчальний посібник. – Хмельницький: Хмельницький національний університет, 2019. 358 с.
14. Bishop, C. M. Pattern Recognition and Machine Learning / C. M. Bishop. – Springer, 2011. – 738 p.
15. Brink, H., Richards, J., Fetherolf, M. Real-World Machine Learning / H. Brink, J. Richards, M. Fetherolf. – Manning Publications, 2016. – 264 p.

Допоміжна

16. Dixon, J. Mastering .NET Machine Learning / J. Dixon. – Packt Publishing Ltd, 2016. – 358 p.
17. Liu, T. F# for C# Developers / T. Liu. – Pearson Education, 2013. – 640 p.
18. Everett B.S. A handbook of statistical analyses using R / B.S. Everett, T/ Hothorn/ - 2 nd edition/ - CRC Press, 2010. 348 p.
19. Marsland S. Machine Learning: An Algorithmic Perspective / S. Marsland. – Chapman and Hall/CRC, 2022. – 457 p.
20. Masood, A. Learning F# Functional Data Structures and Algorithms. / A. Masood. – Packt Publishing Ltd, 2020. – 206 p.
21. Mohri, M., Rostamizadeh, A., Talwalkar A. Foundations of Machine Learning / M. Mohri, A. Rostamizadeh, A. Talwalkar. – The MIT Press, 2022. – 432 p.
22. Mukherjee, S. F# for Machine Learning Essentials / S. Mukherjee. – Packt Publishing Ltd, 2020. – 194 p.
23. Murphy, K. P., Bach, F. Machine Learning: A Probabilistic Perspective / K. P. Murphy, F. Bach. – The MIT Press, 2022. – 1104 p.
24. F# Language Reference. URL: <https://docs.microsoft.com/en-us/dotnet/fsharp/language-reference>.

15. ІНФОРМАЦІЙНІ РЕСУРСИ

1. Модульне середовище для навчання. Доступ до ресурсу: <https://msn.khmnmu.edu.ua/course/view.php?id=7008>
2. Електронна бібліотека ХНУ. Доступ до ресурсу: http://lib.khnu.km.ua/asp/php_f/plage_lib.php.
3. Репозитарій ХНУ. Доступ до ресурсу: <https://elar.khmnmu.edu.ua/home>