

**ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**  
**Факультет інформаційних технологій**  
**Кафедра інженерії програмного забезпечення**



Декан факультету інформаційних технологій  
**Тетяна ГОВОРУЩЕНКО**  
 2024 р.

**СИЛАБУС**

Навчальна дисципліна Мультипарадигмове програмування  
 Освітньо-наукова програма Інженерія програмного забезпечення  
 Рівень вищої освіти Перший (магістерський)  
 Загальна інформація

Позиція	Зміст інформації
Викладач(і)	Яшина Оксана Миколаївна
Профайл викладача	<a href="https://ipz.khmnu.edu.ua/yashyna-o-m/">https://ipz.khmnu.edu.ua/yashyna-o-m/</a>
E-mail викладача(ів)	yashyna.o@khmnu.edu.ua
Контактний телефон	Заповнюється за домовленістю
Сторінка дисципліни в ІСУ	<a href="https://msn.khmnu.edu.ua/course/view.php?id=7008">https://msn.khmnu.edu.ua/course/view.php?id=7008</a>
Навчальний рік	2024-2025
Консультації	Очні: четвер, 3 пара, 1-204 Онлайн: за необхідністю та попередньою домовленістю

**Загальна характеристика дисципліни**

Форма здобуття освіти	Курс	Семестр	Загальний обсяг Кредити ЄКТС	Кількість годин						Курсовий проект	Курсова робота	Форма семестрового контролю	
				Аудиторні заняття				Індивідуальна робота студента	Самостійна робота, в т.ч. ІРС			Залік	Іспит
				Разом	Лекції	Лабораторні роботи	Практичні заняття						
Очна (денна)	1	2	5	54	18	18	18		96			+	

**Анотація навчальної дисципліни**

Дисципліна «Мультипарадигмове програмування» є однією із спеціальних профілюючих дисциплін і тому займає провідне місце у підготовці магістрів інженерії програмного забезпечення. Дисципліна викладається для здобувачів другого (магістерського) рівня вищої освіти денної форми навчання спеціальностей галузі інформаційні технології.

При викладанні дисципліни використовуються активні і творчі форми проведення занять, зокрема, методи проблемного навчання. Дисципліна входить до циклу освітньо-професійної підготовки студента в галузі інженерії програмного забезпечення. Для успішного засвоєння даного курсу здобувач вищої освіти повинен мати навички самостійної роботи з різними джерелами інформації (мережа Інтернет, друковані видання, тощо), умінням застосовувати набуті знання на практиці, володіти умінням узагальнювати інформацію, отриману з різних джерел, коректно представляти результат своєї роботи.

**Мета і завдання дисципліни**

**Мета дисципліни.** Опанування студентами компонентних технологій та інтегрованих середовищ розробки для ефективного виконання конкретних виробничих задач інженерії програмного забезпечення із обґрунтованим вибором парадигм програмування та мультипарадигмових мов програмування.

**Завдання дисципліни:** надати студентам знання і практичні навички практичного вирішення прикладних задач з обґрунтованим вибором мов програмування, в тому числі мультипарадигмових.

Відповідно до Стандарту вищої освіти та освітньої програми дисципліна має забезпечити:

**компетентності:** Здатність проводити дослідження на відповідному рівні. Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення. Здатність розробляти і реалізовувати наукові та/або прикладні проекти у сфері інженерії програмного забезпечення. Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення. Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення. Здатність забезпечувати якість програмного забезпечення.

**програмні результати навчання:** Знати і застосовувати сучасні професійні стандарти і інші нормативно-правові документи з інженерії програмного забезпечення. Розробляти, аналізувати, обґрунтовувати та систематизувати вимоги до програмного забезпечення. Розробляти і оцінювати стратегії проектування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проектних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів. Забезпечувати якість на всіх стадіях життєвого циклу програмного забезпечення, у тому числі з використанням релевантних моделей та методів оцінювання, а також засобів автоматизованого тестування і верифікації програмного забезпечення. Приймати ефективні організаційно-управлінські рішення в умовах невизначеності та зміни вимог, порівнювати альтернативи, оцінювати ризики. Планувати, організовувати та здійснювати тестування, верифікацію та валідацію програмного забезпечення. 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

Студент, який успішно завершив вивчення дисципліни, повинен: Аналізувати, оцінювати і застосовувати на системному рівні сучасні програмні та апаратні платформи для розв'язання складних задач інженерії програмного забезпечення. Обґрунтовано вибирати парадигми і мови програмування для розроблення програмного забезпечення; застосовувати на практиці сучасні засоби розроблення програмного забезпечення. Прогнозувати розвиток програмних систем та інформаційних технологій.

#### Тематичний і календарний план вивчення дисципліни

№ тижня	Тема лекції	Тема практичного заняття	Тема лабораторного заняття	Самостійна робота студента		
				зміст	год	література
1	2	3	4	5	6	7
1-2	Вступ. Основи мультипарадигмового програмування.	Розв'язування задач засобами об'єктно-орієнтованого та функційного програмування F#. Ч. 1	Порівняння мов програмування в аспекті мультипарадигмового програмування. Ч.1	Пророблення теоретичного матеріалу та підготовка до практичної та лабораторної роботи	4	[1,3,4,11, 12,14,29]
3-4	Методика порівняння мультипарадигмових мов програмування	Розв'язування задач засобами об'єктно-орієнтованого та функційного програмування Ч. 2	Порівняння мов програмування в аспекті мультипарадигмового програмування. Ч.2	Пророблення теоретичного матеріалу та підготовка до практичної та лабораторної роботи	6	[1,3,4,11, 12,14,29]
5-6	Програмні та апаратні засоби для розв'язування задач з ПЗ.	Розробка компілятора засобами F#. Ч.1	Здійснення парадигмального аналізу. Ч.1	Пророблення теоретичного матеріалу та підготовка до практичної та лабораторної роботи	8	[1,3,4,11, 12,14,29]
7-8	Мультипарадигмове проектування. Інженерія предметної області в контексті парадигм програмування ..	Розробка компілятора засобами F#. Ч.2	Здійснення парадигмального аналізу. Ч.2	Пророблення теоретичного матеріалу та підготовка до практичної та лабораторної роботи.	8	[1,3,4,11, 12,14,29]

9-10	Використання мультипарадигмового підходу в проектуванні програмного забезпечення.	Розробка парсера засобами F#. Ч.1	Декларативні особливості мови R : використання методу різниці у відмінностях. Ч.1	Пророблення теоретичного матеріалу та підготовка до практичної та лабораторної роботи.	8	[1,3,4,11,12,14,29]
11-12	Мультипарадигмова розробка та аналіз прикладної предметної області	Розробка парсера засобами F#. Ч.2	Декларативні особливості мови R : використання методу різниці у відмінностях. Ч.2	Пророблення теоретичного матеріалу та підготовка до практичної та лабораторної роботи.	8	[1,3,4,11,12,14,29]
13-14	Використання засобів функціонального програмування в мультипарадигмовому аспекті.	Створення розширення за допомогою бібліотек. Ч.1	Декларативні особливості мови R : використання моделі лінійної регресії. Ч.1.	Пророблення теоретичного матеріалу та підготовка до практичної та лабораторної роботи.	8	[2,28,31,34,38]
15-16	Парадигма ймовірнісного програмування	Створення розширення за допомогою бібліотек. Ч.2	Декларативні особливості мови R : використання моделі лінійної регресії Ч.2	Пророблення теоретичного матеріалу та підготовка до практичної та лабораторної роботи.	8	[2,28,31,34,38]
17-18	Перспективи парадигм програмування в умовах глобалізації інформаційних технологій.	Підсумкове заняття	Підсумкове заняття	Пророблення теоретичного матеріалу та підготовка до іспиту	8	[1,3,4,11,12,14,29,34,38]

**Примітка:**\* Послідовність проведення занять визначається розкладом (може не відповідати нумерованим тижням)

#### **Політика дисципліни.**

Організація освітнього процесу з дисципліни відповідає вимогам положень про організаційне і навчально-методичне забезпечення освітнього процесу, освітній програмі та навчальному плану. Студент зобов'язаний відвідувати лекції, лабораторні заняття згідно з розкладом, не запізнюватися на заняття, завдання виконувати відповідно до графіка. Пропущене лабораторне заняття студент зобов'язаний опрацювати самостійно у повному обсязі і відвідувати перед викладачем не пізніше, ніж за тиждень до чергової атестації. До лабораторних занять студент має підготуватися за відповідною темою і проявляти активність. Набуті особою знання з дисципліни або її окремих розділів у неформальній освіті зараховуються відповідно до Положення про порядок перезарахування результатів навчання у ХНУ (<https://khmnu.edu.ua/polozhennya-pro-organizacziyu-osvitnoi-diialnosti/>).

#### **Критерії оцінювання результатів навчання.**

Кожний вид роботи з дисципліни оцінюється за **чотирибальною** шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів навчальної роботи, виконаних і зданих позитивно з врахуванням коефіцієнта вагомості. Вагові коефіцієнти змінюються залежно від структури дисципліни і важливості окремих її видів робіт. При оцінюванні знань студентів використовуються різні засоби контролю, зокрема: усне опитування перед допуском до виконання лабораторної роботи – здійснюється на її початку; засвоєння теоретичного матеріалу з тем перевіряється тестовим контролем; якість виконання, набуття теоретичних знань і практичних навичок перевіряється шляхом захисту кожної лабораторної роботи згідно з робочою програмою дисципліни і робочим навчальним планом.

**Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів денної форми навчання у семестрі за ваговими коефіцієнтами**

Аудиторна робота				Самостійна, індивідуальна робота				Іспит		
<i>Другий семестр</i>										
Лабораторні роботи №:				Практичні роботи №				Контроль:		іспит
1	2	3	4	1	2	3	4	ТК Т1-2	ТК Т 3-4	
ВК*:				0,2				0,2		0,4

*Умовні позначення:* Т – тема дисципліни; ВК – ваговий коефіцієнт; ТК – тестовий контроль.

*Оцінювання тестових завдань.* Тематичний тест для кожного студента складається з двадцяти тестових завдань, кожне з яких оцінюється одним балом. Максимальна сума балів, яку може набрати студент, складає 20. На тестування відводиться 20 хвилин. Тестування проводиться з використанням модульного середовища для навчання MOODLE.

Оцінювання здійснюється за чотирибальною шкалою.

Відповідність набраних балів за тестове завдання оцінці, що виставляється студенту, представлена у нижченаведеній таблиці.

Сума балів за тестове завдання	1–11	12–14	15–18	19-20
Оцінка	2	3	4	5

Підсумкова семестрова оцінка за інституційною шкалою і шкалою ЄКТС встановлюється в автоматизованому режимі після внесення викладачем усіх оцінок до електронного журналу. Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС наведені у таблиці.

**Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС**

Оцінка ЄКТС	Інституційна інтервальна шкала балів	Вітчизняна оцінка, критерії
A	4,75–5,00	5 <b>Відмінно</b> – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навиків
B	4,25–4,74	4 <b>Добре</b> – повне знання навчального матеріалу з кількома незначними помилками
C	3,75–4,24	4 <b>Добре</b> – в загальному правильна відповідь з двома-трьома суттєвими помилками
D	3,25–3,74	3 <b>Задовільно</b> – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією
E	3,00–3,24	3 <b>Задовільно</b> – неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання
FX	2,00–2,99	2 <b>Незадовільно</b> – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни
F	0,00–1,99	2 <b>Незадовільно</b> – необхідна серйозна подальша робота і повторне вивчення дисципліни

**Питання для підсумкового контролю з дисципліни**

1. Поняття про парадигми програмування.
2. Поняття про мультипарадигмові мови програмування.
3. Еволюція парадигм програмування.
4. Сутність, види, класифікація парадигм програмування. Суть мультипарадигмового підходу до програмування.
5. Характеристики та концепції первинних парадигм програмування, імперативних та декларативних (включаючи функціональну та реляційну).
6. Введення поняття процедур та абстракції даних.

7. Дві парадигми програмування.
8. Три парадигми програмування.
9. Чотири парадигми програмування.
10. Функціональна парадигма програмування.
11. Об'єктно-орієнтована парадигма програмування.
12. Логічна парадигма програмування.
13. Рефлексивна парадигма програмування.
14. Основи створення нових мов програмування.
15. Вибір інтерпретатора та компілятора.
16. Розробка інтерпретатора та компілятора.
17. Розширення мов програмування.
18. Вказати методи розширення мов програмування.
19. Використання різних методів для розширення мов програмування.
20. Процедурне та модульне програмування.
21. Опис сучасних мультипарадигмових мов програмування. Їх класифікація та порівняння.
22. Представлення парадигмальних особливостей мов програмування.
23. Результати парадигмального аналізу.
24. Сучасні платформи для використання різних парадигм програмування.
25. Апаратні платформи : аналітика, перспективи розвитку. Вибір оптимального середовища для реалізації проекту.
26. Необхідність кількох парадигм проектування.
27. Визначення мультипарадигмового проектування, аналіз, сімейства, предметні області; прикладний аналіз та аналіз рішень.
28. Сімейства та аналіз спільності. Вимірювання абстракції, точність абстракції. Мультипарадигмове проектування та реалізація.
29. Складні сімейства та включення патернів.
30. Огляд методів мультипарадигмового проектування. Міри складності в проектуванні програмних систем.
31. Проектування з кількома предметними областями в єдиній парадигмі та з різними парадигмами. Кілька парадигм в кожній підобласті.
32. Циклічні підобласті. Операції мультипарадигмового проектування.
33. Переплітання парадигм - аналіз спільності, вимірювання спільності; аналіз мінливості.
34. Взаємозалежні предметні області.
35. Архітектура мультипарадигмового проектування.
36. Рішення які базуються на патернах. Значення ідіом та патернів.
37. Типові патерни - міст, метод шаблонів, стан, уніфікація, синглтон, адаптер.
38. Мультипарадигмове проектування як доповнення патернів
39. Мінливість в часі і просторі. Динамічне зв'язування. Дедуктивна та індуктивна спільність.
40. Сімейства програм. Словник предметної області.
41. Аналіз мінливості. Позитивна та негативна мінливість.
42. Предметна область та діапазон мінливості.
43. Час зв'язування та гнучкість. Ефективність та час зв'язування; варіанти зв'язування.
44. Таблиці мінливості. Графи залежності мінливості.
45. Мультипарадигмовий аналіз прикладної предметної області та області рішень. Сімейства систем, члени сімейства в прикладній області і області рішень.
46. Баланс загального і часткового. Рівні предметної абстракції. Аналіз прикладної області і області рішень. Підобласті в предметному аналізі.
47. Предметний аналіз та багаторазове використання.
48. Модульна організація підобластей. Повторення та ієрархія, структура підобласті; каркас.
49. Аналіз ймовірнісних завдань. Побудова ймовірнісних моделей.
50. Ймовірнісне програмування. Системи ймовірнісних міркувань.
51. Ймовірнісні висновки. Передбачення майбутніх подій. Мультипарадигменна мова ймовірнісного програмування R.
52. Реалізація мови програмування R: синтаксис та функціонал.
53. Виконання простих та складних обчислень.
54. Розподіл випадкових чисел. Інтервальні оцінки. Фрейми даних. Візуалізація даних. Елементи програмування.

### *Методичне забезпечення*

Навчальний процес з дисципліни забезпечений необхідними навчально-методичними розробками в модульному середовищі.

### Рекомендована література

1. Peter Van Roy. Діаграма 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. Проектування 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.
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>.
- 25.

Розробники:



к.т.н., доцент Яшина О.М

Погоджено:

Зав. кафедри ІПЗ



д.ф-м. н., проф. Бедратюк Л.П.

Гарант ОП



к.т.н., доцент Яшина О.М.