

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ



Савенко О.С.
2021р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Мультипарадигмове програмування

Галузь знань 12 «Інформаційні технології»

Спеціальність 121 «Інженерія програмного забезпечення»

Рівень вищої освіти – другий (магістерський)

Освітньо-професійна програма – інженерія програмного забезпечення

Шифр дисципліни – ОПП01

Статус дисципліни – обов'язкова

Факультет – Програмування та комп'ютерних і телекомунікаційних систем

Кафедра – інженерії програмного забезпечення

Форма навчання	Курс	Семестр	Обсяг дисципліни Кредити СКТС	Кількість годин						Форма семестрового контролю			
				Аудиторні заняття						Курсовий проєкт	Курсова робота	Залік	Іспит
				Разом	Лекції	Лабораторні роботи	Практичні заняття	Індивідуальна робота студента	Самостійна робота, в т.ч. ІРС				
Д	1	2	4	54	18	18	18		66				+
Разом ДФН			4	54	18	18	18		66				1

Робоча програма складена на основі освітньо-професійної програми підготовки магістрів

Програма складена  к.т.н, доцент Яшина О.М.

к.т.н, доцент Шестакевич Т.В.

Схвалена на засіданні кафедри Інженерії програмного забезпечення

Протокол № 1 від 31 08 2021 р.

Зав. кафедри ІПЗ  Бедратюк Л.П.

Робоча програма розглянута та схвалена Вченою радою факультету програмування та комп'ютерних і телекомунікаційних систем

Голова Вченої ради  Савенко О.С.

МУЛЬТИПАРАДИГМОВЕ ПРОГРАМУВАННЯ

Тип дисципліни	Обов'язкова
Рівень вищої освіти	Другий (магістерський)
Мова навчання	Українська
Семестр	другий
Форми навчання, для яких викладається дисципліна	Денна

Результати навчання Відповідно до Стандарту вищої освіти дисципліна має забезпечити *компетентності*: Здатність до абстрактного мислення, аналізу та синтезу. Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення. Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення. 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. Здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення. Здатність виконувати проектування програмних систем та їх складових компонентів із використанням поглиблених знань з оптимізації, реінжинірингу програмних систем, управління проектами та якістю з вибором парадигм програмування та застосування машинного навчання. **програмні результати навчання:** 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. 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.

Зміст навчальної дисципліни. Мультипарадигмове програмування: поняття, порівняння, програмні та апаратні платформи для розв'язування задач з ІІЗ. Сутність, види, класифікація парадигм програмування. Суть мультипарадигмового підходу до програмування. Мультипарадигмове проектування. Інженерія предметної області в контексті парадигм програмування. Використання мультипарадигмового підходу в проектуванні програмного забезпечення. Мультипарадигмова розробка та аналіз прикладної предметної області. Функціональна та ймовірнісна парадигми програмування. Використання засобів функціонального програмування в мультипарадигмовому аспекті. Парадигма ймовірнісного програмування. Перспективи парадигм програмування в умовах глобалізації інформаційних технологій.

Запланована навчальна діяльність: лекції – 18 год., лабораторні заняття – 18 год., практичні заняття – 18; самостійна робота – 66 год., разом – 120 год.

Методи навчання: методи проблемного викладання, словесні, наочні (лекції); пояснювально-ілюстративні, дослідницькі, частково-пошукові (лабораторні заняття, практичні заняття), дослідницькі, частково-пошукові (самостійна робота: індивідуальні завдання).

Форми і методи оцінювання результатів навчання: усне опитування, захисти лабораторних та практичних робіт, письмовий іспит.

Вид семестрового контролю: іспит.

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, 2010.
2. Elad Shalom A Review of Programming Paradigms Throughout the History: With a Suggestion Toward a Future Approach / USA: Independently published, 2015. – 345 p.

3. Peter Van Roy. Диаграмма с результатами сравнения более 30-ти парадигм программирования. <https://www.info.ucl.ac.be/~pvr/paradigmsDIAGRAMeng108.pdf>.
4. Модульне середовище для навчання MOODLE. Доступ до ресурсу: <https://msn.khnu.km.ua>.
5. Електронна бібліотека університету. Доступ до ресурсу: http://lib.khnu.km.ua/asp/php_f/p1age_lib.php.

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

3. ПОЯСНЮВАЛЬНА ЗАПИСКА

Дисципліна «Мультипарадигмове програмування» є однією із фахових дисциплін і займає провідне місце у підготовці фахівців освітнього рівня «магістр» за спеціальністю 121 «Інженерія програмного забезпечення» за освітньо-професійною програмою «Інженерія програмного забезпечення».

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.

Відповідно до Стандарту вищої освіти дисципліна має забезпечити

компетентності: ЗК-1. Здатність до абстрактного мислення, аналізу та синтезу. ФК-1. Здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення. ФК5. Здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення. ФК-7. 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. ФК-10 Здатність виконувати проектування програмних систем та їх складових компонентів 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-7 Аналізувати, оцінювати і застосовувати на системному рівні сучасні програмні та апаратні платформи для розв'язання складних задач інженерії програмного забезпечення. ПРН-9 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.

4. СТРУКТУРА ЗАЛКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ

Назва теми	Кількість годин, відведених на:			
	Лекції	Лаб. роботи	Практичні роботи	СР С
Тема 1. Мультипарадигмове програмування: поняття, порівняння, програмні та апаратні платформи для розв'язування задач з ІІЗ.	6	8	8	16
Тема 2. Мультипарадигмове проектування.	6	4	4	16
Тема 3. Функціональна та ймовірнісна парадигми програмування.	4	2	2	18
Тема 4. Перспективи розвитку парадигм програмування в контексті інформаційних технологій.	2	4	4	16
Разом	18	18	18	66

5. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

5.1. Зміст лекційного курсу

№ лекції	Перелік тем лекцій, їх анотації	К-сть годин
1	2	3
1	<p>Тема1. Мультипарадигмове програмування: поняття, порівняння, програмні та апаратні платформи для розв’язування задач з ІІЗ.</p> <p><i>Лекція 1. Вступ. Основи мультипарадигмового програмування.</i> 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.: [1,3,4,11,12,14,29]</p>	2
2	<p>Тема1. Мультипарадигмове програмування: поняття, порівняння, програмні та апаратні платформи для розв’язування задач з ІІЗ.</p> <p><i>Лекція 2. Методика порівняння мультипарадигмових мов програмування</i> Опис сучасних мультипарадигмових мов програмування. Їх класифікація та порівняння. Представлення парадигмальних особливостей мов програмування. Результати парадигмального аналізу. Літ.: [2,13,14,19,20,29]</p>	2
3.	<p>Тема1. Мультипарадигмове програмування: поняття, порівняння, програмні та апаратні платформи для розв’язування задач з ІІЗ.</p> <p><i>Лекція 3. Програмні та апаратні засоби для розв’язування задач з ІІЗ.</i> Сучасні платформи для використання різних парадигм програмування. Апаратні платформи : аналітика, перспективи розвитку. Вибір оптимального середовища для реалізації проекту. Літ.: [2,14,15,20,23]</p>	
4	<p>Тема 2.Мультипарадигмове проектування.</p> <p><i>Лекція 4 Мультипарадигмове проектування. Інженерія предметної області в контексті парадигм програмування</i> 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.: [2,14,15,20,23]</p>	2

5	<p>Тема 2.Мультипарадигмове проектування. <i>Лекція 5. Використання мультипарадигмового підходу в проектуванні програмного забезпечення.</i></p> <p>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</p>	2
---	--	---

	<p>на патернах. Значення ідіом та патернів. Типові патерни - міст, метод шаблонів, стан, уніфікація, синглтон, адаптер. Мультипарадигмове проектування як доповнення патернів. Літ.: [2,14,15,20,23]</p>	
--	---	--

6	<p>Тема 2. Мультипарадигмове проектування. <i>Лекція 6. Мультипарадигмова розробка та аналіз прикладної предметної області</i></p> <p>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 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 підобластей; каркас. Літ.: [6,13,14,15,39]</p>	2
---	---	---

7	<p>Тема 3.Функціональна та ймовірнісна парадигми програмування. <i>Лекція 7. Використання засобів функціонального програмування в мультипарадигмовому аспекті.</i></p> <p>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 лямбдавиразів. Розмічені об'єднання. Використання розмічених об'єднань для створення деревовидних структур. Методи і властивості розмічених об'єднань. Записи. Типи Lazy. 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.: [6,7,8,9,10,22]</p>	2
---	---	---

8	<p>Тема 3. Функціональна та ймовірнісна парадигми програмування. <i>Лекція 8. Парадигма ймовірнісного програмування</i> Аналіз ймовірнісних завдань. Побудова ймовірнісних моделей. Ймовірнісне програмування. Системи ймовірнісних міркувань. Ймовірнісні висновки. Передбачення майбутніх подій. Мультипарадигменна мова ймовірнісного програмування R. Реалізація мови програмування R: синтаксис та функціонал. Виконання простих та складних обчислень. Розподіл випадкових чисел. Інтервальні оцінки. Фрейми даних. Візуалізація даних. Елементи програмування. Приклади програм. Літ.: [2,28,31,34,38]</p>	2
9.	<p>Тема 4. Перспективи розвитку парадигм програмування в контексті інформаційних технологій. <i>Лекція 9. Перспективи парадигм програмування в умовах глобалізації інформаційних технологій.</i> Зміна умов експлуатації сучасних програмних систем та проєктів: специфіка, вимоги, терміни. Використання методів прогнозування розвитку в умовах інформаційних технологій. Літ.: [2,28,31,34,38]</p>	2
	Разом за семестр	18

5.2. Зміст лабораторних занять

№ з/п	Тема лабораторного заняття	Кількість годин
1	<p>Лабораторна робота №1. Частина 1 Порівняння мов програмування в аспекті мультипарадигмового програмування Літ.: [1,3,4,11,12,14,29]</p>	2
2	<p>Лабораторна робота №1. Частина 2 Порівняння мов програмування в аспекті мультипарадигмового програмування Літ.: [1,3,4,11,12,14,29]</p>	2
3	<p>Лабораторна робота №2. Частина 1 Здійснення парадигмального аналізу Літ.: [2,13,14,19,20,29]</p>	2
4	<p>Лабораторна робота №2. Частина 2 Здійснення парадигмального аналізу Літ.: [2,13,14,19,20,29]</p>	2
5	<p>Лабораторна робота №3. Частина 1 Декларативні особливості мови R : використання методу різниці у відмінностях. Літ.: [6,7,8,9,10,22,31,34]</p>	2

6	Лабораторна робота №3. Частина 2 Декларативні особливості мови R : використання методу різниці у відмінностях. Літ.: [2,28,31,34,38]	2
7	Лабораторна робота №4. Частина 1 Декларативні особливості мови R : використання моделі лінійної регресії Літ.: [2,28,31,34,38]	2
8	Лабораторна робота №4. Частина 2 Декларативні особливості мови R : використання моделі лінійної регресії Літ.: [2,28,31,34,38]	2
9	Підсумкове заняття	2
	Разом за семестр	18

5.3. Зміст практичних занять

№ з/п	Тема практичного заняття	Кількість годин
1	Практична робота №1. Частина 1 Розв'язування задач засобами об'єктно-орієнтованого та функційного програмування Літ.: [6,7,8,9,10,22,31,34]	2
2	Практична робота №1. Частина 2 Розв'язування задач засобами об'єктно-орієнтованого та функційного програмування Літ.: [6,7,8,9,10,22,31,34]	2
3	Практична робота №2. Частина 1 Розробка компілятора засобами F# Літ.: [6,7,8,9,10,22,31,34]	2
4	Практична робота №2. Частина 2 Розробка компілятора засобами F# Літ.: [6,7,8,9,10,22]	2
5	Практична робота №3. Частина 1 Розробка парсера засобами F#. Літ.: [6,7,8,9,10,22,31,34]	2
6	Практична робота №3. Частина 2 Розробка парсера засобами F#. Літ.: [6,7,8,9,10,22]	2
7	Практична робота №4. Частина 1 Створення розширення за допомогою бібліотек. Літ.: [6,7,8,9,10,22,31,34]	2

8	Практична робота №4. Частина2 Створення розширення за допомогою бібліотек. Літ.: [6,7,8,9,10,22,31,34]	2
9	Підсумкове заняття	2
	Разом за семестр	18

5.5 Зміст самостійної роботи

Самостійна робота студентів *денної* форми навчання полягає у систематичному опрацюванні програмного матеріалу, підготовці до виконання і захисту лабораторних робіт, тестування з теоретичного матеріалу, виконанні індивідуальних завдань, в тому числі курсового проекту, тощо.

Номер тижня	Зміст самостійної роботи	Кількість годин
1	2	3
1	Опрацювання лекційного матеріалу, підготовка до виконання лабораторної роботи № 1.	2
2	Опрацювання лекційного матеріалу Підготовка до захисту лабораторної роботи №1	2
3	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи №2. Опрацювання довідкової літератури.	2
4	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи № 2	4
5	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи № 3. Опрацювання довідкової літератури.	4
6	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи № 3. Опрацювання довідкової літератури.	4
7	Опрацювання лекційного матеріалу. Підготовка до виконання лабораторної роботи № 4. Опрацювання довідкової літератури.	4
8	Опрацювання лекційного матеріалу та довідкової літератури. Підготовка до ТК1.	4
9	Опрацювання лекційного матеріалу. Підготовка до захисту лабораторної роботи № 4. Опрацювання довідкової літератури.	4
10	Опрацювання лекційного матеріалу. Підготовка до виконання практичної роботи № 1. Опрацювання довідкової літератури.	4
11	Опрацювання лекційного матеріалу. Підготовка до виконання практичної роботи №2. Опрацювання довідкової літератури.	4

12	Опрацювання лекційного матеріалу. Підготовка до захисту практичної роботи № 1. Опрацювання довідкової літератури.	4
13	Опрацювання лекційного матеріалу. Підготовка до захисту практичної роботи № 2. Опрацювання довідкової літератури.	4
14	Опрацювання лекційного матеріалу. Підготовка до виконання практичної роботи № 3. Опрацювання довідкової літератури.	4
15	Опрацювання лекційного матеріалу. Підготовка до захисту практичної роботи №3. Опрацювання довідкової літератури.	4
16	Опрацювання лекційного матеріалу. Підготовка до виконання практичної роботи №4. Опрацювання довідкової літератури.	4
17	Опрацювання лекційного матеріалу. Підготовка до захисту практичної роботи №4. Підготовка до задачі ТК2.	4
18	Опрацювання запитань для самоповторення. Підготовка до складання іспиту.	4
Разом		66

6. МЕТОДИ НАВЧАННЯ

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних методів: методи проблемного викладання, словесні, наочні (лекції); пояснювально-ілюстративні, дослідницькі, частково-пошукові (лабораторні заняття, практичні заняття), дослідницькі, частково-пошукові (самостійна робота: індивідуальні завдання). Всі заняття проводяться з використанням інформаційних технологій і мають за мету – 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

7. ФОРМИ І МЕТОДИ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Поточний контроль здійснюється під час лекційних та лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочим планом дисципліни. Семестровий контроль проводиться у формі іспиту. При виведенні остаточної оцінки враховуються результати поточного контролю та письмового іспиту

Використовуються наступні методи поточного контролю:

- усне опитування студентів перед допуском до лабораторного заняття;
- допуск до захисту лабораторних робіт;
- захист лабораторних робіт;
- тестування та перевірка працездатності створеного у ході виконання лабораторної роботи програмного, технічного та іншого забезпечення;

Підсумкова оцінка виставляється з урахуванням коефіцієнтів вагомості.

Студент, який не допущений до захисту або не захистив лабораторні роботи, вважається невстигаючим.

Процес оцінювання підготовленості студента можна розділити на етапи:

Перший етап оцінювання направлений на визначення знань інформаційного мінімуму. Якщо студент твердо засвоїв визначену навчальним планом суму формальних знань, то це означає, що він вміє використати їх при вирішенні різних питань при проектуванні виробів із шкіри, вміє розширити їх. Перед вивченням дисципліни, як правило, проводиться вхідний контроль знань з дисциплін, що їй передують і забезпечують. При цьому необхідно встановити рівні та критерії сформованості знань щодо змісту навчальних елементів. Такими рівнями є:

Ознайомчо-орієнтовний (ОО) – особа має орієнтовне уявлення щодо понять, які вивчаються, здатна: відтворювати формулювання визначень поняття мобільних платформ; орієнтуватись в методиках розробки програмного продукту; знати теоретичні основи створення додатків для мобільних платформ.

Понятійно-аналітичний (ПА) – особа має чітке уявлення щодо навчального об'єкту, здатна здійснювати смислове виділення, пояснення вибору конструктивних рішень щодо створення програмного продукту, методик їх проектування, їх обґрунтування на основі аналізу існуючих. Може чітко визначити структуру програм при їх проектуванні в залежності від їх використання, тобто здатна перенести раніше засвоєні знання на типові ситуації.

Продуктивно-синтетичний (ПС) – особа має глибоке розуміння щодо навчального об'єкту, здатна здійснювати синтез, генерувати нові ідеї та уявлення, переносити раніше засвоєні знання на нетипові, нестандартні ситуації. Тобто на цьому рівні студент повинен на основі теоретичних знань вміти проектувати різні види програмного продукту за найбільш поширеними методиками, вносити свої пропозиції щодо удосконалення способів проектування та одержання додатків для мобільних платформ; створювати конкурентоспроможні програмні додатки на світовому ринку.

При оцінюванні знань студентів використовуються різні засоби контролю, зокрема:

-допуск до виконання лабораторної роботи здійснюється на її початку усним опитуванням студента;
-якість виконання лабораторної роботи оцінюється шляхом перевірки виконаної лабораторної роботи у робочому файлі та в робочому зошиті;

-набуття теоретичних знань і практичних навичок перевіряється шляхом захисту кожної лабораторної роботи;

-засвоєння теоретичного матеріалу перевіряється тестовим контролем.

Оцінка, яка виставляється за лабораторне заняття, складається з таких елементів: усне опитування студентів перед допуском до виконання лабораторної роботи; знання теоретичного матеріалу з теми; якість оформлення результатів лабораторної роботи у зошиті та в робочому файлі; вміння студента обґрунтовувати прийняті рішення та робити висновки; своєчасний захист лабораторної роботи. Термін захисту лабораторної роботи вважається своєчасним, якщо студент захистив її на наступному після виконання роботи занятті. За несвоєчасний захист лабораторної роботи (без поважної причини) захист лабораторної роботи оцінюється у три бали.

Пропущене з поважної причини лабораторне заняття студент повинен відпрацювати в лабораторіях кафедри в установленій викладачем термін.

Кожний вид роботи оцінюється за чотирибальною шкалою. Підсумкова оцінка з дисципліни виставляється як середньозважена з усіх видів робіт. Вагові коефіцієнти (ВК) видів робіт дисципліни приведено в таблицях 1-2.

Таблиця 1 - 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				0,2		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

Оцінка роботи студента за традиційною системою

Оцінка *"відмінно"* виставляється студенту, який вільно володіє матеріалом, чітко, грамотно, логічно, повно і послідовно висвітлює питання, демонструє знання не лише лекційного матеріалу, але й результати самостійного опрацювання літератури, вміння висвітлювати власні міркування та робити висновки.

Найвища оцінка ставиться студенту, який показав всебічне та глибоке знання навчально-програмного матеріалу, розбирається в поняттях, принципах та методах аналізу вимог до ПЗ, а також: який засвоїв основну і знайомий з додатковою літературою, засвоїв основні поняття дисципліни; дав повні і точні відповіді на всі запитання з поясненнями і висновками; проявив творчі здібності в розумінні, викладенні і використанні навчального матеріалу.

Оцінка *"добре"* виставляється студенту, який добре володіє навчальним матеріалом, грамотно і логічно викладає його в усній чи письмовій формі, допускаючи при цьому не більше одного-двох недоліків чи незначних помилок, Обмежується викладом основних теоретичних положень, не володіє матеріалом додаткової літератури, хоча загалом має тверді знання.

"Задовільно" отримує студент, що показав знання основного матеріалу в обсязі, необхідному для подальшого навчання (основні поняття, принципи, закономірності), а також: знайомий з основною літературою; при відповідях на запитання відчуває себе невпевнено, допускає незначні помилки; при допусканні погрішностей у відповідях володіє знаннями для їх виправлення під керівництвом викладача.

Оцінка *"незадовільно"* виставляється студенту, який демонструє не зовсім повні знання курсу навчальної дисципліни чи її окремої теми, допускає при викладанні навчального матеріалу суттєві помилки, не орієнтується у лекційному матеріалі та матеріалі основної і додаткової літератури.

"Незадовільно" ставиться студентові, який має пробіли в знаннях основного матеріалу, а також: допустив принципові помилки, не розуміє основних концептуальних положень дисципліни; на запитання не дає задовільних відповідей, не справляється з рішенням задач; не може продовжувати навчання без додаткових занять з дисципліни.

Якщо студент отримав негативну оцінку, то він має перездати її в установленому порядку, але обов'язково до терміну наступного контролю. У випадку, коли студент не виконав індивідуальний план з дисципліни у заплановані терміни без поважних причин, то під час відпрацювання заборгованості при позитивній відповіді йому виставляється оцінка „задовільно”.

Підсумкова семестрова оцінка за національною шкалою і шкалою ЄКТС встановлюється в автоматизованому режимі після внесення усіх оцінок до електронного журналу. Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС наведені у табл. 2.

Залік виставляється, якщо середньозважений бал, який отримав студент з дисципліни, знаходиться в межах від 3,00 до 5,00 балів. При цьому за вітчизняною шкалою ставиться «зараховано», а за шкалою ЄКТС – буквене позначення оцінки, що відповідає набраній студентом кількості балів.

Перехід від вітчизняної шкали оцінювання до європейської (ECTS) представлено у таблиці 2:

Таблиця 2 - Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ECTS	Бали	Вітчизняна система	
A	4,75-5,00	5	ВІДМІННО - глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навичок
B	4,25-4,74	4	ДОБРЕ - повне знання навчального матеріалу з кількома незначними помилками
C	3,75-4,24	4	ДОБРЕ - в загальному правильна відповідь з двома-трьома суттєвими помилками
D	3,25-3,74	3	ЗАДОВІЛЬНО - неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією
E	3,00-3,24	3	ЗАДОВІЛЬНО - неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання
FX	2,00-2,99	2	НЕЗАДОВІЛЬНО - безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни
F	0,00-1,99	2	НЕЗАДОВІЛЬНО - необхідна серйозна подальша робота і повторне вивчення дисципліни

8. ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ РЕЗУЛЬТАТІВ НАВЧАННЯ

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. Мультипарадигмова 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.

9. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Навчальний процес з дисципліни повністю і в достатній кількості забезпечений необхідною навчально-методичною літературою в модульному середовищі.

10. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

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, 2010.
- Aleksei Sedunov Kotlin In-Depth [Vol-I]: A Comprehensive Guide to Modern Multi-Paradigm Language / USA: BPB Publications, 2019. – 348 p.
2. Bedir Tekinerdogan Multi-Paradigm Modelling Approaches for Cyber-Physical Systems / USA: Academic Press, 2020. – 314 p.
3. Chris Smith Programming F# 3.0 / USA: O'Reilly Media, 2012. – 476 p.
4. Tomas Petricek, Jon Skeet Real-World Functional Programming: With Examples in F# and C# / USA: Manning Publications, 2012. – 500 p.
5. Joseph Fernandez Learn F#: Functional Programming / ROI Press, 2016. – 152 p.
6. Wiki Books F# Programming / USA: opensource-books, 2014. – 436 p.
7. Michael Metcalf, John Reid, Reid Metcalf The F Programming Language / USA: Oxford University Press, 2016. – 344 p.
8. Elad Shalom A Review of Programming Paradigms Throughout the History: With a Suggestion Toward a Future Approach / USA: Independently published, 2015. – 345 p.
9. Daniel Holden Build Your Own Lisp / USA: ebook publishing, 2014. – 276 p.
10. О. Перевозчикова Інформаційні системи і структури даних. – К. : К-М Академія. – 2011. – 288 с.
11. Д. Сідоров, І. Казак Основи інженерних розрахунків на ПЕОМ. Програмування алгоритмічною мовою Фортран. – В: ЦНЛ. – 186 с.

12. Говорущенко Т. О. Методологія оцінювання достатності інформації для визначення якості програмного забезпечення: монографія. Хмельницький: Хмельницький національний університет, 2017. – 310 с.
18. Бевз О. М. Проектування програмних засобів систем управління: навчальний посібник / О. М. Бевз, В. М. Папінов, Ю. А. Скидан. – Вінниця: ВНТУ, 2010. – 125 с.
19. Лаврищева Е. М. Software Engineering комп'ютерних систем. Парадигми, технології та CASEзасоби програмування. – Київ: Наукова думка, 2013. – 283 с.
20. Говорущенко Т. О. Аналіз, дослідження та оцінювання програмних систем: навчальний посібник. – Хмельницький: Хмельницький національний університет, 2019. 358 с.
21. Bishop, C. M. Pattern Recognition and Machine Learning / C. M. Bishop. – Springer, 2011. – 738 p.
22. Brink, H., Richards, J., Fetherolf, M. Real-World Machine Learning / H. Brink, J. Richards, M. Fetherolf. – Manning Publications, 2016. – 264 p.
23. Dixon, J. Mastering .NET Machine Learning / J. Dixon. – Packt Publishing Ltd, 2016. – 358 p.
24. Liu, T. F# for C# Developers / T. Liu. – Pearson Education, 2013. – 640 p.
25. Everett B.S. A handbook of statistical analyses using R / B.S. Everett, T. Hothorn / - 2 nd edition / - CRC Press, 2010. 348 p.

Допоміжна

29. Marsland S. Machine Learning: An Algorithmic Perspective / S. Marsland. – Chapman and Hall/CRC, 2014. – 457 p.
30. Masood, A. Learning F# Functional Data Structures and Algorithms. / A. Masood. – Packt Publishing Ltd, 2015. – 206 p.
31. Mohri, M., Rostamizadeh, A., Talwalkar A. Foundations of Machine Learning / M. Mohri, A. Rostamizadeh, A. Talwalkar. – The MIT Press, 2012. – 432 p.
32. Mukherjee, S. F# for Machine Learning Essentials / S. Mukherjee. – Packt Publishing Ltd, 2016. – 194 p.
33. Murphy, K. P., Bach, F. Machine Learning: A Probabilistic Perspective / K. P. Murphy, F. Bach. – The MIT Press, 2012. – 1104 p.
34. Shalev-Shwartz, S., Ben-David, S. Understanding Machine Learning: From Theory to Algorithms / S. Shalev-Shwartz, S. Ben-David. – Cambridge University Press, 2014. – 449 p.
35. F# Language Reference [Електронний ресурс] : [Інтернет-портал]. – Електронні дані. – [© Microsoft, 2018]. – Режим доступу: <https://docs.microsoft.com/en-us/dotnet/fsharp/language-reference/>

11. ІНФОРМАЦІЙНІ РЕСУРСИ

1. Модульне середовище для навчання. Доступ до ресурсу: <https://msn.khnu.km.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: <http://elar.khnu.km.ua/jspui/?locale=uk>.