



МУЛЬТИПАРАДИГМОВІ МОВИ ПРОГРАМУВАННЯ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Третій (доктор філософії)
Галузь знань	12 Інформаційні технології
Спеціальність	122 Комп'ютерні науки
Освітня програма	КОМП'ЮТЕРНІ НАУКИ
Статус дисципліни	Нормативна
Форма навчання	очна(денна)
Рік підготовки, семестр	2 курс, весняний семестр
Обсяг дисципліни	2 кредити (60 годин)
Семестровий контроль/ контрольні заходи	Залік
Розклад занять	<i>Перший та другий тиждень: Лекція/практична робота, четвер, 10-25, on-line</i>
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: <i>к.т.н. Кравцов Григорій Олексійович, контактні дані: hryhoriy.kravtsov@gmail.com</i> Практичні : <i>к.т.н. Кравцов Григорій Олексійович, контактні дані: hryhoriy.kravtsov@gmail.com</i>
Розміщення курсу	Посилання на дистанційний ресурс (Moodle, Googleclassroom, тощо):

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Дисципліна «Мультипарадигмові мови програмування» (ІВ2-3) є нормативною дисципліною навчального плану підготовки докторів філософії з спеціальності «Комп'ютерні науки» і грає важливу роль у підготовці фахівців.

Метою навчальної дисципліни є формування у аспірантів глибокого розуміння функціонального програмування, його основних концепцій та застосування у сучасному програмуванні. Курс прагне забезпечити студентів навичками, які дозволять їм ефективно використовувати функціональні мови програмування, такі як Scala, і зрозуміти математичні основи, що лежать в основі програмувальних парадигм.

Метою кредитного модуля є формування у аспірантів загальних та спеціальних (фахових, предметних) компетентностей:

- ЗК01 – здатність до абстрактного мислення, аналізу та синтезу;
- ЗК03 — здатність працювати в міжнародному контексті.

- ЗК04 – здатність розробляти проекти та управляти ними;
- СК02. Здатність усно і письмово презентувати та обговорювати результати наукових досліджень з комп'ютерних наук та/або інноваційних розробок державною та іноземною (англійською або іншими) мовами, глибоке розуміння іншомовних;
- СК03 – здатність створювати і застосовувати сучасні інформаційні технології, архітектури і спеціалізоване програмне забезпечення у науковій та освітній діяльності, керувати інформаційними ресурсами, інформаційними системами та цифровими сервісами;
- СК05 – здатність розвивати теоретичні засади, створювати моделі інформаційних технологій, проектувати та створювати інформаційні системи і цифрові сервіси та їх прототипи;
- СК06 – здатність застосовувати сучасні методи дослідження, синтезу, проектування інформаційних систем і технологій у науковій та науково-педагогічній діяльності;

Програмні результати навчання:

Згідно з вимогами програми навчальної дисципліни аспіранти після засвоєння кредитного модуля мають продемонструвати такі результати навчання:

- ПРН01. Мати передові концептуальні та методологічні знання з комп'ютерних наук і на межі предметних галузей, а також дослідницькі навички, достатні для проведення наукових і прикладних досліджень на рівні останніх світових досягнень з відповідного напрямку, отримання нових знань та/або здійснення інновацій.
- ПРН04. Розробляти та досліджувати концептуальні, математичні і комп'ютерні моделі процесів і систем, ефективно використовувати їх для отримання нових знань та/або створення інноваційних продуктів у комп'ютерних науках та дотичних міждисциплінарних напрямках.
- ПРН06. Застосовувати сучасні інструменти і технології пошуку, оброблення та аналізу інформації, зокрема, статистичні методи аналізу даних великого обсягу та/або складної структури, спеціалізовані бази даних та інформаційні системи.
- ПРН07. Розробляти та реалізовувати наукові та/або інноваційні інженерні проекти, які дають можливість переосмислити наявне та створити нове цілісне знання та/або професійну практику і розв'язувати значущі наукові та технологічні проблеми комп'ютерної науки з дотриманням норм академічної етики і врахуванням соціальних, економічних, екологічних та правових аспектів.
- ПРН09. Вивчати, узагальнювати та впроваджувати в навчальний процес інновації комп'ютерних наук.
- ПРН10. Відшуковувати, оцінювати та критично аналізувати інформацію щодо поточного стану та трендів розвитку, інструментів та методів досліджень, наукових та інноваційних проектів з комп'ютерних наук.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Пререквізити: здатність застосовувати знання у практичних ситуаціях, навички використання інформаційних і комунікаційних технологій. Цей курс базується на таких забезпечуючих дисциплінах: тенденції розвитку інформаційних систем та технологій, сучасні концепції створення інтелектуальних систем, проектування інформаційних систем.

Постреквізити: Перелік напрямків діяльності, що забезпечуються: педагогічна практика, захист дисертаційної роботи.

3. Зміст навчальної дисципліни

Модуль 1. Вступ до функціонального програмування.

Тема 1.1. Функції і дані. Незмінність даних. Трейти, класи, абстрактні класи. Option.

Тема 1.2. Співставлення патернів. Відкладені обчислення. Рекурсія та оптимізація хвостової рекурсії. For-вирази та синтаксичне зв'язування.

Тема 1.3. Управління виконанням програми.

Модуль 2. Монади та алгебраїчні типи

Тема 1.4. Алгебраїчні типи даних, типи-суми і типи-добутки.

Тема 1.5. Поняття групи, моноїда, функтора та монади. Список як монада.

Тема 1.6. Option та Either як монади. Future

Модуль 3. Функціональні патерни та спеціальні теми

Тема 1.7. Тайпкласи як засіб композиції функцій. Власні тайпкласи.

Тема 1.8. Програма як перетворення станів. Стан як композиція функцій. Монада State.

4. Навчальні матеріали та ресурси

Базова

1. Mark C. Lewis and Lisa Lacher, Introduction to Programming and Problem-Solving Using Scala, 2 ed., 2016.
Англійська версія: <http://www.programmingusingscala.net/>
Розділи 2-8, 10, 15
2. Martin Odersky et al., Programming in Scala, 2020.
Англійська версія: <https://www.artima.com/pins1ed/>
Українська версія: <http://ac2epsilon.github.io/TRANS/SCALA/pis3-adoc/>
Розділи 1-13, 15-21, 28
3. Scala 3 Book.
Посилання: <https://docs.scala-lang.org/scala3/book/introduction.html>
4. Навчальна платформа Udey.com, Scala & Functional Programming Essentials | Rock the JVM
– Режим доступу до ресурсу:
<https://www.udemy.com/course/rock-the-jvm-scala-for-beginners/>
5. Навчальна платформа Pluralsight.com, Scala Language: The Big Picture [Електронний ресурс].
– Режим доступу до ресурсу:
<https://app.pluralsight.com/library/courses/scala-language-big-picture/table-of-contents>
6. Навчальна платформа Pluralsight.com, Thinking Functionally in Scala 2
– Режим доступу до ресурсу:
<https://app.pluralsight.com/library/courses/scala-thinking-functionally/table-of-contents>

Додаткова

1. Документація Scala.
Посилання: <https://docs.scala-lang.org/>
2. Bartosz Milewski, Category Theory for Programmers.
Посилання:
<https://github.com/hmemcpy/milewski-ctfp-pdf/releases/download/v1.3.0/category-theory-for#programmers.pdf>

3. F# for Fun and Profit.

Посилання: <https://fsharpforfunandprofit.com/site-contents/>

Це сайт присвячений програмуванню на F#, але тут є багато корисних статей загальної тематики. Рекомендуємо звернути увагу на дві:

- Functional Programming Design Patterns.
- Railway Oriented Programming

Навчальний контент

5. Методика опанування навчальної дисципліни(освітнього компонента)

№ з/п	
	Розділ 1. Вступ до функціонального програмування.
1	<p>Лекція 1. Функції і дані. Незмінність даних. Типи даних в Scala Концепція імутабельності в контексті функціонального програмування. Переваги незмінних структур даних. Класи і трейти. Введення в функції як основні будівельні блоки функціонального програмування. Використання типу Option для безпечного представлення відсутності значення. Розгляд функцій як структури першого класу в мовах програмування. Поняття і використання функцій вищих порядків. Література: 1-3 Самостійна робота: створення проекту мовою програмування Scala за допомогою інструменту SBT бібліотеки scalatest</p>
2	<p>Лекція 2. Співставлення патернів. Відкладені обчислення. Рекурсія та оптимізація хвостової рекурсії. For-вирази та синтаксичне зв'язування. Вивчення механізму співставлення патернів, який дозволяє зручно розбирати дані на складові. Огляд рекурсії як фундаментального інструменту у функціональному програмуванні. Основи лінивих обчислень та їх відмінності від жадібних обчислень. Застосування синтаксичного зв'язування для створення читабельного та елегантного коду. Література: 1-3 Самостійна робота: створення case класу та трейту. 3 прикладами лінивих обчислень та співставлення зі зразком</p>
3	<p>Лекція 3. Управління виконанням програми. Зупинка виконання програми. Аварійне завершення роботи функції. Використання функціональних патернів для управління виконанням програми. Введення в концепції управління потоками виконання програми. Розгляд синхронного та асинхронного виконання коду. Приклади функціональних патернів, таких як монади, для управління станом і виконанням. Література: 1-3 Самостійна робота: за допомогою бібліотеки показати приклад аварійної зупинки виконання функції. Монадичної функції. Зупинка виконання роботи програми з використанням типу Try.</p>
	Розділ 2. Монади та алгебраїчні типи
4	<p>Лекція 4. Алгебраїчні типи даних, типи-суми і типи-добутки. Ознайомлення з основними концепціями алгебраїчних типів даних. Розгляд прикладів алгебраїчних типів даних у Scala. Вивчення типів-сум, їх синтаксису та використання у Scala. Аналіз прикладів типів-сум, таких як Either та Option. Огляд типів-добутків та їх реалізації у Scala.</p>

	Література: 2-4 Самостійна робота: Створення деревовидного алгебраїчного типу даних
5	Лекція 5. Поняття групи, моноїда, функтора та монади. Список як монада. Визначення та основні властивості груп у математиці. Застосування концепції груп у програмуванні. Введення в моноїди та їх асоціативність. Приклади моноїдів у Scala, такі як операції зі строками та числами. Означення функторів та їх використання у Scala. Розгляд map методу та його застосування до колекцій. Визначення монади у контексті функціонального програмування. Розгляд flatMap та unit операцій, які є основою для монад. Література: 1-3 Самостійна робота: Створення простого списку.
6	Лекція 6. Option та Either як монади. Future. Розгляд методів map, flatMap, getOrElse та інших, які роблять Option та Either монадою. Аналіз списку в Scala як монади. Використання списків для створення послідовних обчислень з використанням монадичних операцій. Вивчення монад як засобу управління побічними ефектами та асинхронністю. Література: 1-4. Самостійна робота: Отримання асинхронних результатів виконання функції за допомогою Future.
	Розділ 3. Функціональні патерни та спеціальні теми
7	Лекція 7. Тайпкласи як засіб композиції функцій. Власні тайпкласи. Введення в бібліотеку тайпкласів Scala Cats. Вивчення аплікативних функторів та їх відмінностей від звичайних функторів. Огляд процесу створення та використання власних тайпкласів у Scala. Приклади, які демонструють розробку та інтеграцію власних тайпкласів. Література: 3-6. Самостійна робота: створення власного тайпкласу для маніпуляції текстовими даними String
8	Лекція 8. Програма як перетворення станів. Стан як композиція функцій. Монада State. Програма як перетворення станів. Ознайомлення з поняттям стану як результату виконання функцій. Аналіз композиції функцій для моделювання складних станів. Введення в монаду State як засіб управління станом у функціональних програмах. Розгляд використання монади State для створення чистих функцій, що змінюють стан. Приклади використання монади State у реальних Scala-проектах. Література: 3-6. Самостійна робота: зміна стану даних за допомогою монади State та Future

Практичні заняття

№ з/п	Назва практичної роботи (комп'ютерного практикуму)	Кількість ауд. годин
1	Практична робота 1. Створення проекту за допомогою SBT та підключення бібліотек таких, як Scalatest та Cats. Література: 1,2.	2
2	Практична робота 2. Запуск тестів і демонстрація можливостей зміни значень за допомогою debug	2

	Література: 1-3.	
3	Практична робота 3. Цикли. Рекурсія і демонстрація помилки <i>Stack Overflow</i> . Література: 1-3.	2
4	Практична робота 4. Читання файлів і аварійна зупинка виконання програми. Обробка помилок. Література: 2-3.	2
5	Практична робота 5. Створення проекту за допомогою <i>Play Framework</i> . Робота <i>Web-сервісів</i> . Література: 1,2.	2
6	Практична робота 6. Використання можливостей <i>Debug</i> для пошуку помилок у веб-сервісі. Парсинг даних. Література: 1-2, 4-6.	2
7	Практична робота 7. Підключення до бази даних <i>Mongo</i> та збереження і видача результатів. Література: 2, 5-6.	2
8	Практична робота 8. Стримінг даних з бази <i>Mongo</i> . Асинхронне виконання коду на <i>Web-server</i> . Література: 1-2, 5-6.	2
9	Практична робота 9. Підключення до кластеру <i>Apache kafka</i> та отримання даних Література: 2-5.	2
10	Практична робота 10. Отримання даних з файлу наукового формату <i>HDFS</i> та відображення за допомогою <i>HTML</i> на <i>Play Framework</i> Література: 2.	

6. Самостійна робота аспіранта

№ з/п	Назви тем і питань, що виносяться на самостійне опрацювання та посилання на навчальну літературу	Кількість годин СРС
1	2	3
1	Створення проекту за допомогою <i>SBT</i> та публікація його в системі <i>GitLab</i> [1-3]	4
2	Парсинг <i>JSON</i> з файлу та написання тестів на це [2-4]	4
3	Парсинг даних з <i>XML</i> файлу [2-4]	4
4	Підключення до бази даних <i>SQL</i> та отримання даних із таблиці [2-5]	4
5	Огляд фреймворків для створення веб-сервісів [2-5]	4
6	Обробка даних, які отримуються з потоку <i>kafka</i> [2-5]	4
7	Створення веб сервісу з підтримкою <i>Web Socket</i> [3-6]	6

8	Написання програми симуляції ходів коня на шаховій дошці [2-5]	6
9	Розміщення веб-сервісу з відображення даних файлів HDF5 в публічній хмарній системі [3-6]	6

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Відвідування занять

Відвідування лекцій переконливо рекомендується, але штрафних санкцій за пропуски лекцій не передбачено. Відвідування занять комп'ютерного практикуму необхідно в обсязі, достатньому для виконання вимог викладача щодо виконання і своєчасної здачі практичних робіт та індивідуального завдання.

Пропущені контрольні заходи

Лабораторні роботи можна здавати у відведений за розкладом час лабораторних занять, як до, так і після встановленого терміну здачі конкретної роботи. Додаткові години для складання індивідуального завдання призначаються викладачем в межах часу лабораторних занять. За відсутності поважних причин пропуску (медична довідка тощо) штрафні бали не нараховуються.

Процедура оскарження результатів контрольних заходів

Здобувачі мають можливість підняти будь-яке питання, яке стосується процедури контрольних заходів та очікувати, що воно буде розглянуто згідно із наперед визначеними процедурами.

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Поточний контроль: виконання індивідуального завдання на проектування інформаційних систем, виконання комп'ютерних практикумів.

Календарний контроль: проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

Семестровий контроль: залік.

Умови допуску до семестрового контролю: мінімально позитивна оцінка за індивідуальне завдання /зарахування усіх практичних робіт/ семестровий рейтинг більше 30 балів.

Рейтинг аспіранта з дисципліни складається з двох складових: стартової – призначена для оцінювання заходів поточного контролю впродовж семестру та залікової – призначена для оцінювання окремих запитань (завдань) на екзамені і формується з балів, що він отримує за:

- 1) індивідуальне завдання на проектування інформаційної системи;
- 2) виконання комп'ютерних практикумів;
- 3) відповідь на заліку.

1. Індивідуальне завдання

Ваговий бал – 25. Кожний аспірант виконує індивідуальне завдання щодо розробки інформаційної системи (програмного забезпечення) в рамках свого дисертаційного дослідження, яке передбачає використання всієї низки програмно-технічного інструментарію, що вивчається в рамках курсу.

2. Робота на комп'ютерних практичних заняттях

Ваговий бал – 5. Максимальна кількість балів на всіх практичних заняттях дорівнює $5 \cdot 9 = 45$ балів.

3. Відповідь на заліку

Кількість балів по відповіді на кожне питання визначається викладачем з врахуванням складності питання та якості відповіді. Максимальна кількість балів 30.

Штрафні та заохочувальні бали за:

- виконання завдань із удосконалення дидактичних матеріалів з дисципліни надається від 2 до 5 заохочувальних балів.

Розрахунок шкали (R) рейтингу

Сума вагових балів контрольних заходів протягом семестру складає:

$RC = 70 + 30 = 100$ балів.

Для отримання аспірантом відповідних оцінок (ECTS та традиційних) його рейтингова оцінка *R* переводиться згідно з таблицею відповідності рейтингових балів оцінкам за університетською шкалою.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація з дисципліни (освітнього компонента)

- можливість зарахування сертифікатів проходження дистанційних чи онлайн курсів за тематикою дисципліни «Мультипардигмові мови програмування»;
- інша інформація для аспірантів щодо особливостей опанування навчальної дисципліни:

Застосовуються стратегії активного і колективного навчання, які визначаються наступними методами і технологіями:

- 1) кредитно-модульна технологія навчання;
- 2) особистісно-орієнтовані (розвиваючі) технології, засновані на активних формах і методах навчання («аналіз ситуацій» ділові, імітаційні ігри, дискусія, експрес-конференція, навчальні дебати.);
- 3) інформаційно-комунікаційні технології, що забезпечують проблемно-дослідницький характер процесу навчання та активізацію самостійної роботи аспірантів (електронні презентації для лекційних занять, використання аудіо-, відео-підтримки навчальних занять, розробка і застосування на основі комп'ютерних і мультимедійних засобів творчих завдань, доповнення традиційних навчальних занять засобами взаємодії на основі мережевих комунікаційних можливостей).

Робочу програму навчальної дисципліни (силабус):

Складено к.т.н. Кравцовим Григорієм Олексійовичем.

Ухвалено: Вченою радою ІПМЕ ім. Г.Є. Пухова НАН України (протокол №10 від 26.09.2024 р.)