



# ФУНКЦІОНАЛЬНІ МОВИ ПРОГРАМУВАННЯ

## Робоча програма навчальної дисципліни (Силабус)

### Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Третій (доктор філософії)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>122 Комп'ютерні науки</i>
Освітня програма	<i>КОМП'ЮТЕРНІ НАУКИ</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>очна(денна)</i>
Рік підготовки, семестр	<i>3 курс, весняний семестр</i>
Обсяг дисципліни	<i>2 кредити (60 годин)</i>
Семестровий контроль/ контрольні заходи	<i>Залік</i>
Розклад занять	<i>Перший та другий тиждень: Лекція/практична робота, четвер, 10-25, <u>он- line</u></i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Лектор: к.т.н. Кравцов Григорій Олексійович, контактні дані: <a href="mailto:hryhoriy.kravtsov@gmail.com">hryhoriy.kravtsov@gmail.com</a> Практичні : к.т.н. Кравцов Григорій Олексійович, контактні дані: <a href="mailto:hryhoriy.kravtsov@gmail.com">hryhoriy.kravtsov@gmail.com</a></i>
Розміщення курсу	

### Програма навчальної дисципліни

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

Дисципліна «Функціональні мови програмування» (ІВ2-1) є нормативною дисципліною навчального плану підготовки докторів філософії зі спеціальності «Комп'ютерні науки» і грає важливу роль у підготовці фахівців.

Метою навчальної дисципліни є формування в 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 в 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 й 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. Відшукувати, 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.*

## **Модуль 2. Монади та алгебраїчні типи**

Тема 1.3. Алгебраїчні типи даних, типи-суми та типи-добутки. Поняття групи, моноїда, функтора та монади. Список як монада.

## **Модуль 3. Функціональні патерни та спеціальні теми**

Тема 1.4. Тайпкласи як засіб композиції функцій. Власні тайпкласи. 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 Udemy.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

5. Методика опанування навчальної дисципліни(освітнього компонента)

№ з/п	
	<b>Розділ 1. Вступ до функціонального програмування.</b>
1	<p><b>Лекція 1. Функції та дані. Незмінність даних. Типи даних в Scala</b>  <i>Концепція імутабельності в контексті функціонального програмування. Переваги незмінних структур даних. Класи та 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.</i>                      Література: 1-3                      Самостійна робота: створення проєкту мовою програмування Scala за допомогою інструменту SBT бібліотеки scalatest</p>
2	<p><b>Лекція 2. Співставлення патернів. Відкладені обчислення. Рекурсія та оптимізація хвостової рекурсії. For-вирази та синтаксичне зв'язування.</b>  <i>Вивчення механізму зіставлення патернів, який дозволяє зручно розбирати дані на складники. Огляд рекурсії як фундаментального інструменту у функціональному програмуванні. Основи лінійних обчислень та їх відмінності від жадібних обчислень. Застосування синтаксичного зв'язування для створення читабельного та елегантного коду.</i>                      Література: 1-3                      Самостійна робота: створення case класу та трейту. 3 прикладами лінійних обчислень та зіставлення зі зразком</p>
	<b>Розділ 2. Монади та алгебраїчні типи</b>
3	<p><b>Лекція 3. Алгебраїчні типи даних, типи-суми та типи-добутки. Поняття групи, моноїда, функтора та монади. Список як монада.</b>  <i>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. 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 тар методу та його застосування до колекцій. Визначення монади у контексті функціонального програмування. Розгляд flatMap та unit операцій, які є основою для монад.</i>                      Література: 2-4                      Самостійна робота: Створення деревоподібної структури за допомогою алгебраїчного типу даних</p>
	<b>Розділ 3. Функціональні патерни та спеціальні теми</b>
4	<p><b>Лекція 4. Тайпкласи як засіб композиції функцій. Власні тайпкласи. Програма як перетворення станів. Стан як композиція функцій. Монада State</b>  <i>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. 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-проєктах.</i>                      Література: 3-6.                      Самостійна робота: створення власного тайпкласу для маніпуляції текстовими даними String</p>

## Практичні заняття

№ з/п	Назва практичної роботи (комп'ютерного практикуму)	Кількість ауд. годин
1	<i>Практична робота 1. Створення проєкту за допомогою SBT та підключення бібліотек таких, як Scalatest та Cats.</i> Література: 1,2.	2
2	<i>Практична робота 2. Цикли. Рекурсія і демонстрація помилки Stack Overflow.</i> Література: 1-3.	2
3	<i>Практична робота 3. Читання файлів і аварійна зупинка виконання програми. Обробка помилок.</i> Література: 2-3.	2
4	<i>Практична робота 4. Створення проєкту за допомогою Play Framework. Робота Web-сервісів.</i> Література: 1,2.	2
5	<i>Практична робота 5. Підключення до бази даних Mongo та збереження і видача результатів.</i> Література: 2, 5-6.	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 проєкту за допомогою SBT та публікація його в системі GitLab [1-3]	4
2	Парсинг JSON з файлу та написання тестів на це [2-4]	4
3	Парсинг даних з XML файлу [2-4]	4
4	Підключення до бази даних SQL 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 kafka [2-5]	4
7	Створення веб сервісу з підтримкою Web Socket [3-6]	6
8	Написання програми симуляції ходів коня на шаховій дошці [2-5]	6
9	Розміщення веб-сервісу з відображення даних файлів HDFS 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 порушувати 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 понад 30 балів.

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

1) індивідуальне завдання на проектування інформаційної системи;

2) виконання комп'ютерних практикумів;

3) відповідь на екзамені.

#### **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 з удосконалення дидактичних матеріалів з дисципліни надається від 2 до 5 заохочувальних балів.

#### **Розрахунок шкали (R) рейтингу**

Сума вагових балів контрольних заходів протягом семестру складає:

$$RC = 70 + 30 = 100 \text{ балів.}$$

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
Менш як 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 Функціональні мови програмування»;

- інша інформація для аспірантів щодо особливостей опанування навчальної дисципліни:

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

1) кредитно-модульна технологія навчання;

2) особистісно-орієнтовані (розвивальні) 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-, відеопідтримки 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.)