

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ
КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ПРОГРАМНИХ СИСТЕМ**

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчальної роботи

_____ Кашпур О.Ф.

«___» _____ 2019 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
НЕКЛАСИЧНІ ЛОГІКИ ТА ЇХ
ЗАСТОСУВАННЯ В РОЗРОБЦІ
ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ
для студентів**

галузь знань	12 Інформаційні технології
спеціальність	121 Інженерія програмного забезпечення
освітній рівень	магістр
освітня програма	Програмне забезпечення систем
спеціалізація	Програмне забезпечення систем
вид дисципліни	вибіркова

Форма навчання	денна
Навчальний рік	2021/2022
Семестр	3
Кількість кредитів ECTS	3
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	залік

Викладач: **д. ф.-м. н., професор Провотар О.І.** (лекції)

Пролонговано: на 20__/20__ н. р. _____ (_____) «__» 20__ р.

на 20__/20__ н. р. _____ (_____) «__» 20__ р.

Розробник: Провотар Олександр Іванович, д. ф.-м. н., професор, завідувач кафедри інтелектуальних програмних систем.

ЗАТВЕРДЖЕНО

Завідувач кафедри інтелектуальних програмних систем

_____ О.І. Провотар

Протокол № __ від «__» _____ 2019 р.

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

Протокол від «__» _____ 2019 року №__

Голова науково-методичної комісії _____ Л.Л. Омельчук

«__» _____ 2019 року

Затверджено вченою радою факультету комп'ютерних наук та кібернетики

Протокол від «__» _____ 2019 року №__

Голова вченої ради факультету _____ А.В. Анісімов

1. Метою дисципліни «Некласичні логіки та їх застосування в розробці програмного забезпечення» є розширення знань з теорії категорій та нечіткої логіки для побудови відповідних математичних моделей подання знань.

2. Попередні вимоги до опанування або вибору навчальної дисципліни. Для засвоєння курсу необхідні знання основ елементарної математики, дискретної математики, алгебри, математичної логіки та теорії алгоритмів, елементів категорного аналізу та нечітких логік.

1. Знати: основи елементарної математики, дискретної математики, алгебри, математичної логіки та теорії алгоритмів, елементів категорного аналізу та нечітких логік.
2. Вміти: застосовувати на практиці інструментальні програмні засоби проектування та розробки нечіткого програмного забезпечення, проектувати та розробляти категорні моделі алгоритмізації та подання знань.
3. Володіти навичками: проектувати та розробляти нечіткі моделі подання знань, застосовувати на практиці категорні засоби дослідження обчислювальності базових конструкцій побудови алгоритмів.

3. Анотація навчальної дисципліни. Навчальна дисципліна «Некласичні логіки та їх застосування в розробці програмного забезпечення» є складовою програми підготовки фахівців за другим (магістерським) рівнем вищої освіти галузі знань 12 Інформаційні технології зі спеціальності 121 Інженерія програмного забезпечення, освітньо-наукової програми «Програмне забезпечення систем».

Дана дисципліна є навчальною дисципліною вільного вибору студентів в рамках блоку спеціалізації «Програмне забезпечення систем».

Дисципліна є продовженням дисциплін «Нечіткі логіки», «Елементи категорного аналізу» та «Алгебро-автоматні методи проектування програмного забезпечення».

Дисципліна викладається у 3 семестрі в **обсязі – 90 год. (3 кредити ECTS)**, зокрема: лекції – 28 год., самостійна робота – 60 год., консультації – 2 год. В курсі передбачено 2 змістовних частини та 2 контрольні роботи. Завершується дисципліна – **заліком**.

В результаті вивчення навчальної дисципліни студент повинен:

знати: основні поняття та визначення категорної та нечіткої логік, принципи побудови та дослідження систем нечіткого логічного виведення, інструментальні засоби інтеграції категорних та нечітких моделей в програмний продукт.

вміти: використовувати інструментальні засоби інтеграції категорних та нечітких моделей в програмний продукт, будувати та досліджувати категорні та нечіткі моделі подання знань.

4. Завдання (навчальні цілі). Набуття знань, умінь та навичок (компетентностей) на рівні основ розробки категорних та нечітких моделей і їх інтеграції в програмний продукт, відповідно до кваліфікації фахівця з інформаційних технологій. Зокрема, розвивати:

- Здатність до абстрактного мислення, аналізу та синтезу (ЗК01).
- Здатність проведення теоретичних та прикладних досліджень на відповідному рівні (ЗК03).
- Здатність удосконалювати свої навички на основі аналізу попереднього досвіду (ЗК06).
- Здатність приймати обґрунтовані рішення (ЗК08).
- Здатність аналізувати предметні області, формувати, аналізувати та моделювати вимоги до програмного забезпечення (СК01).

- Здатність проектувати програмне забезпечення, включаючи проведення моделювання його архітектури, поведінки та процесів функціонування окремих підсистем і модулів (СК03).
- Здатність систематизувати професійні знання щодо створення і супроводження програмного забезпечення (СК07).
- Здатність застосовувати і розвивати фундаментальні і міждисциплінарні знання для успішного розв'язання наукових завдань інженерії програмного забезпечення (СК10).
- Здатність до алгоритмічного та логічного мислення (СК11.1).

5. Результати навчання за дисципліною.

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
РН1.1	Знати основні поняття і визначення категорної та нечіткої логік.	Лекції.	Контрольна робота (тест) 1, 60% правильних відповідей.	15%
РН1.2	Знати основні принципи розробки моделей подання знань на основі категорних та нечітких моделей.	Лекції.	Контрольна робота (тест) 1, 2, 60% правильних відповідей.	10%
РН1.3	Знати основні методи та інструментальні засоби розробки програмного забезпечення на основі категорних та нечітких моделей.	Лекції.	Контрольна робота (тест) 2, 60% правильних відповідей.	15%
РН2.1	Вміти застосовувати на практиці інструментальні засоби розробки програмного забезпечення на основі категорних та нечітких моделей.	Самостійна робота.	Тест, 60% правильних відповідей.	24%
РН3.1	Вміти будувати категорні та нечіткі моделі подання знань	Самостійна робота.	Захист самостійної роботи (проекту).	20%
РН4.1	Організовувати свою самостійну роботу для досягнення результату	Самостійна робота.	Захист самостійної роботи (проекту).	8%
РН4.2	Відповідально ставитися до виконуваних робіт,	Самостійна робота.	Захист самостійної роботи (проекту).	8%

	нести відповідальність за їх якість			
--	-------------------------------------	--	--	--

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання.

Результати навчання дисципліни	РН1.1	РН1.2	РН1.3	РН2.1	РН3.1	РН4.1	РН4.2
	Програмні результати навчання						
ПРН01. Знати і системно застосовувати методи аналізу та моделювання прикладної області, виявлення інформаційних потреб і збору вихідних даних для проектування програмного забезпечення.	+	+	+	+		+	+
ПРН03. Знати і застосовувати базові концепції і методології моделювання інформаційних процесів.		+		+		+	+
ПРН08. Проводити аналітичне дослідження параметрів функціонування програмних систем для їх валідації та верифікації, а також проводити аналіз обраних методів, засобів автоматизованого проектування та реалізації програмного забезпечення.			+	+	+	+	+
ПРН13. Оформляти результати досліджень у вигляді статей у наукових виданнях та тез доповідей на науково-технічних конференціях.						+	+
ПРН14. Пояснити, аналізувати, цілеспрямовано шукати і обирати необхідні для вирішення фахових наукових і прикладних задач інформаційно-довідкові та науково-технічні ресурси і джерела знань з урахуванням сучасних досягнень науки і техніки.						+	+
ПРН16.1. Володіння методами та технологіями нейромереж, неklasичних логік, квантових обчислень, кластерних розрахунків, категорного аналізу для розв'язання завдань інженерії програмного забезпечення.	+	+		+	+	+	+

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

Семестрове оцінювання:

1. Контрольна робота 1: РН 1.1, РН 1.2 – **20/12 балів.**
2. Контрольна робота 2: РН 1.2, РН1.3 – **20/12 балів.**
3. Самостійна робота 1 (проект): РН 2.1, РН3.1, РН 4.1, РН4.2 – **30/18 балів.**
4. Самостійна робота 2 (проект): РН 2.1, РН3.1, РН 4.1, РН4.2 – **20/18 балів.**

7.2 Організація оцінювання:

Терміни проведення форм оцінювання:

1. Контрольна робота: до 7 тижня семестру.
2. Контрольна робота: до 15 тижня семестру.
3. Самостійна робота 1 (проект): до 7 тижня семестру.
4. Самостійна робота 2 (проект): до 14 тижня семестру.

Студент має право на одне перескладання кожної контрольної роботи із можливістю отримання максимально 80% початково визначених за цю контрольну роботу балів. Термін перескладання визначається викладачем.

У випадку відсутності студента з поважних причин відпрацювання та перездачі контрольних робіт здійснюються у відповідності до «Положення про порядок оцінювання знань студентів при кредитно-модульній системі організації навчального процесу» від 1 жовтня 2010 року.

У разі неякісного виконання лабораторної роботи, викладач має право не зарахувати лабораторну роботу, або знизити за неї бали.

Студент має право здавати лабораторні роботи після закінчення визначеного для них терміну, але з втратою одного балу за кожен тиждень, який пройшов з моменту закінчення терміну її здачі.

Здобувач освіти може бути недопущений до підсумкового оцінювання, якщо під час семестру він:

- 1) не досяг мінімального порогового рівня оцінки тих результатів навчання, які не можуть бути оцінені під час підсумкового контролю;
- 2) набрав кількість балів, що є недостатньою для отримання позитивної оцінки навіть у випадку досягнення ним на підсумковому контролі максимально можливого результату.

7.3 Шкала відповідності оцінок

Зараховано / Passed	60-100
Не зараховано / Fail	0-59

8. Структура навчальної дисципліни. Тематичний план лекцій.

№ лекції	Назва лекції	Кількість годин	
		Лекції	Самостійна робота
Частина 1. Категорна логіка.			
1	Тема 1. Будова топоса. Мономорфізми як порівнювачі. Образи морфізмів. Екстенціональність і двозначність.	4	4
2			
3	Тема 2. Логіка в топосі. Поява логіки в топосі. Пропозиціональні числення. Булеві алгебри.	2	4
4	Тема 3. Алгебра підоб'єктів. Алгебра підоб'єктів – доповнення, перетин, об'єднання. $\text{Sub}(d)$ – гратка.	2	4
5	Тема 4 Булеві топоси. Перегляд принципу екстенціональності.	2	4
6	Тема 5. Арифметика. Натурально-числові об'єкти. Примітивна рекурсія.	2	4
7	Тема 6. Опис основних категорних конструкцій мовою Haskell.	4	6
8			
Контрольна робота 1			2
Контроль за підсумками самостійної роботи 1			2
Всього по частині 1		16	30
Частина 2. Нечіткі логіки та моделі.			

9	Тема 7. Системи нечіткого логічного виведення.	2	5
10	Тема 8. Системи нечіткого логічного виведення в прийнятті рішень.	2	5
11	Тема 9. Методи побудови нечітких моделей.	2	5
12	Тема 10. Достовірність в системах нечіткого логічного виведення.	4	5
13			
14	Тема 11. Інтеграція нечітких моделей в програмний продукт мовою Python.	2	6
Контрольна робота 2			2
Контроль за підсумками самостійної роботи 2			2
Всього по частині 2		12	30
Консультація		2	
ВСЬОГО		28	60

Загальний обсяг – **90** год., в тому числі:

Лекції – **28** год.

Самостійна робота – **60** год.

Консультації – **2** год.

Теми, винесені на самостійне вивчення.

1. Цілий та раціональний об'єкти.
2. Визначення мономорфності та епіморфності через елементи.
3. Побудова контролерів.
4. Розробка експертних систем з нечіткими моделями подання знань.
5. Методи прогнозування на основі нечітких моделей подання знань.
6. Базові засоби мови Python для роботи з нечіткими моделями.

Самостійні роботи.

1. Експертні системи з нечіткими моделями подання знань.
2. Видобування знань за допомогою наближених множин.

9. Рекомендовані джерела.

Основні:

1. Д. Рутковская, М. Пилиньский, Л. Рутковский. Нейронные сети, генетические алгоритмы и нечеткие системы. М.: Телеком, 2006. – 382 с.
2. J. Leski. Systemy neuronowo-rozmyte. Warszawa: Naukowo-Techniczne, 2008. – 690 с.
3. Zadeh L.A. Fuzzy sets as a basis for a theory of possibility //Fuzzy Sets ana Systems, 1978, N1, p. 3–28.
4. Голдблатт Р. Топосы. Категорный анализ логики. – М. – Мир. – 1983.
5. Джонстон П. Теория топосов – М. – Мир. – 1978.
6. А.Г. Куц. Топосы. Задания и методические указания / Омск. Ун-т, 1989, 35 с.
7. Bartosz Milewski. Category Theory for Programmers. Version v1.0.0-g41e0fc3. October 21, 2018.

8. Лутц М. Изучаем Python, 4-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2011.

Додаткові:

1. Статті в журналах.