

**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
ПРОГРАМНО-ОРІЄНТОВАНИ ЛОГІКИ
для студентів**

галузь знань	12 Інформаційні технології
спеціальність	121 Інженерія програмного забезпечення
освітній рівень	магістр
освітня програма	Програмне забезпечення систем
вид дисципліни	обов'язкова

Форма навчання	денна
Навчальний рік	2020/2021
Семестр	2
Кількість кредитів ECTS	3
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	залік

Викладач: **д. ф.-м. н., професор Кривий С.Л.** (лекції).

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

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

Розробник: Кривий Сергій Лук'янович д. ф.-м. н., професор, професор 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. Мета дисципліни. Метою і завданням навчальної дисципліни «Програмно-орієнтовані логіки» є поглиблене вивчення математичної логіки, а саме програмно-орієнтованих логік. В прикладному аспекті, апарат математичної логіки необхідний для адекватного моделювання різноманітних предметних областей, створення сучасних програмних та інформаційних систем.

2. Попередні вимоги до опанування навчальної дисципліни. Для успішного вивчення дисципліни «Програмно-орієнтовані логіки» студенти повинні відповідати наступним вимогам:

1. Успішне опанування курсів:
 1. Дискретна математика.
 2. Загальна алгебра.
 3. Теорія алгоритмів та математична логіка.
 4. Об'єктно-орієнтоване програмування.
 5. Нечіткі логіки.
2. Знання:
 1. Основних концепцій процедурного, об'єктно-орієнтованого, функціонального та логічного програмування.
 2. Процедурного та об'єктно-орієнтованого програмування мовою C++.
 3. Об'єктно-орієнтованого програмування мовою Java.
3. Вміти:
 1. Виконувати аналіз проблеми, що виникає.
 2. Будувати математичні моделі відповідних предметних областей.
 3. Створювати специфікації та виконувати верифікацію моделей, що виникли в результаті аналізу проблеми.
 4. Розв'язувати задачі перевірки специфікацій на моделях.
 5. Програмувати у процедурному, функціональному, логічному та об'єктно-орієнтованому стилях.
4. Володіти:
 1. Базовими навичками використання інтегрованих середовищ розробки програмного забезпечення CLion/ IntelliJIDEA/ Eclipse/ NetBeans.
 2. Англійською мовою на рівні не нижче Upper Intermediate.

3. Анотація дисципліни. Навчальна дисципліна «Програмно-орієнтовані логіки» 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 нормативною навчальною дисципліною в рамках освітньої програми «Програмне забезпечення систем». Викладається у 2 семестрі в **обсязі – 90 год. (3 кредити ECTS)**, зокрема: лекції – 28 год., самостійна робота – 60 год., консультації – 2 год. В курсі передбачено 2 змістовні частини та 2 контрольні роботи. Завершується дисципліна – **заліком**.

Дисципліна «Програмно орієнтовані логіки» виступає базовою для дисциплін «Прикладні логіки та елементи квантових обчислень» в рамках освітньої програми «Програмне забезпечення систем».

4. Завдання (навчальні цілі). Основними завданнями дисципліни «Програмно орієнтовані логіки» є набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень в області програмно-орієнтованих логік відповідно до освітньої кваліфікації магістр з інженерії програмного забезпечення. 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).
- Вміння планувати і проводити наукові дослідження, готувати результати наукових робіт з інженерії програмного забезпечення до оприлюднення (СК09).

5. Результати навчання за дисципліною.

Результат навчання (1. знати; 2. вміння; 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 алгоритми теорії автоматів, їх складність та варіації (базовий рівень).	Лекції, самостійна робота.	Контрольні роботи №1 та №2, поточне оцінювання завдань винесених на самостійну роботу 1, 2.	10%
РН1.2	Знати основні поняття та концепції загальної алгебри (напівгрупи, групи, ґратки, багатоосновні алгебри тощо).	Лекції, самостійна робота.		5%
РН1.3	Знати основні етапи повного циклу проектування, їх задачі та методи розв'язання.	Лекції, самостійна робота.		35%
РН2.1	Вміти проектувати та розробляти на основі побудованих математичних моделей алгоритми розв'язання поставлених задач, виконувати аналіз складності побудованих алгоритмів.	Самостійна робота.	Поточне оцінювання завдань винесених на самостійну роботу 1, 2.	20%
РН3.1	Консультуватися з викладачем стосовно	Самостійна робота, години		5%

	питань що виникають у ході вивчення теоретичного матеріалу.	консультацій.		
РН3.2	Обговорювати з колегами та викладачем проблемі питання що виникають у ході виконання лабораторних робіт.	Самостійна робота.		5%
РН3.3	Чітко та послідовно обґрунтовувати власні проектні рішення в рамках виконання лабораторних робіт.	Самостійна робота.		5%
РН4.1	Закріплення та поглиблення набутих на лекціях теоретичних знань проектування та розробки програм.	Самостійна робота, опрацювання рекомендованих інформаційних джерел.		5%
РН4.2	Закріплення та поглиблення набутих на лабораторних заняттях практичних навичок проектування та розробки програм.	Самостійна робота, реалізація рекомендованих завдань.		5%
РН4.3	Сумлінне і вчасне виконання завдань самостійних робіт.	Самостійна робота.		5%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання.

Результати навчання дисципліни	РН1.1	РН1.2	РН1.3	РН2.1	РН3.1	РН3.2	РН3.3	РН4.1	РН4.2	РН4.3
	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.																				
---------------------------------	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

7. Схема формування оцінки.

7.1. Форми оцінювання студентів

Семестрове оцінювання:

1. Контрольна робота 1: РН1.1, РН1.2, РН3.1 – **15 балів/9 балів.**
2. Контрольна робота 2: РН1.2, РН1.3, РН3.1 – **15 балів/9 балів.**
3. Самостійна робота 1: РН1.1, РН1.2, РН2.1, РН3.1, РН3.2, РН3.3, РН4.1, РН4.2, РН4.3 – **10 балів/6 балів.**
4. Самостійна робота 2: РН1.2, РН1.3, РН2.1, РН3.1, РН3.2, РН3.3, РН4.1, РН4.2, РН4.3 – **20 балів/12 балів.**

Підсумкове оцінювання (у формі заліку):

- Залікові бали визначаються як сума оцінок/балів за всіма успішно оціненими результатами навчання передбачених даною програмою.
- Оцінки нижче від мінімального порогового рівня не додаються.
- Мінімальний пороговий рівень для кожного окремого компонента становить 50% від максимально можливої кількості балів.
- Мінімальний пороговий рівень для сумарної оцінки за всіма компонентами становить 60% від максимально можливої кількості балів.

7.2 Організація оцінювання.

Терміни проведення форм оцінювання:

1. Контрольна робота 1: до 7 тижня семестру.
2. Контрольна робота 2: до 14 тижня семестру.
3. Самостійна робота 1: до 3 тижня семестру.
4. Самостійна робота 2: до 5 тижня семестру.

Студенти мають право на одне перескладання кожної контрольної роботи у визначений викладачем термін.

У випадку відсутності студентів 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.

7.2. 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
<p>Частина 1. Класичні логіки та їх дедуктивні системи. Метод резолюцій та проблема уніфікації. термів в теоріях першого порядку. Транзиційні системи та їх специфікації. Методи перевірки виконуваності специфікацій засобами класичних логік.</p>			
1	Тема 1. Числення висловлювань. Аксиоматика та правила виведення. Основні властивості: несуперечність, повнота в широкому та вузькому сенсі, Дедукція.	2	4
2	Тема 2. Методи перевірки виконуваності формул ЧВ: Куайна, редукції до абсурду, резолюцій, семантичного табло.	2	4
3	Тема 3. Числення предикатів. Аксиоматика та правила виведення. Уніфікація та метод резолюцій.	2	4
4	Тема 4. Транзиційні системи. Композиції ТС та специфікації властивостей станів та переходів ТС. Гомоморфізми та еквівалентності ТС. Факторизація ТС.	4	4
5			
6	Тема 5. Синхронний добуток ТС та його властивості. Моделювання синхронного добутку ТС мережами Петрі та автоматами.	4	8
7			
8	Тема 6. Поняття про некласичні логіки та їх зв'язок з класичними логіками. Приклади застосування до специфікації властивостей ТС та автоматів.	2	4
Контрольна робота № 1			2
Всього по частині 1		16	30
<p>Частина 2. Некласичні логіки. Модальні та темпоральні логіки та їх застосування до верифікації реактивних та функціональних систем. Методи семантичного табло та метод перевірки на моделі для некласичних логік.</p>			
9	Тема 7. ЛТЛ. Аксиоматика та правила виведення. Метод семантичного табло для ЛТЛ. Застосування до специфікації властивостей систем.	4	8
10			
11	Тема 8. Моделі Кріпке та узагальнені автомати Бюхі. Трансляція моделей та ЛТЛ-формул в ці автомати. Приклади побудови автоматів для конкретних властивостей.	2	5
12	Тема 9. Логіки розгалуженого часу. Аксиоматика та правила виведення. Виразність. Застосування цих логік до специфікації властивостей паралельних та розподілених систем.	2	5

13	Тема 10. Методи перевірки виконуваності формул логік розгалуженого часу на моделі. Зв'язок з автоматами.	2	5
14	Тема 11. Застосування специфікацій формулами логік розгалуженого часу до реальних систем. Автоматизовані системи перевірки виконуваності властивостей. Системи ESTELLE, RAISE та інші. Підсумки.	2	5
Контрольна робота № 2			2
Всього по частині 2		12	30
Консультації		2	
ВСЬОГО		28	60

Загальний обсяг годин – **90** год., у тому числі:

Лекцій – **28** год.

Самостійна робота – **60** год.

Консультації – **2** год.

Контрольні запитання до частини 1.

1. Основні поняття аксіоматичної теорії. Дедукція і індукція. Приклади.
2. Принципи побудови дедуктивної логічної системи. Числення. Правила виведення.
3. Зв'язок логіки висловлювань і булевої алгебри. Числення висловлювань.
4. Несуперечність та повнота в синтаксичному і семантичному сенсах.
5. Логіка предикатів. Аксіоматика та правила виведення. Властивості.
6. Методи резолюцій та семантичного табло. Алгоритми перевірки виконуваності. Уніфікація та метчинг. Алгоритми уніфікації.
7. Модальна пропозиційна логіка. Оператори загальності та існування. Властивості.
8. Семантика, Реляційна модель Кріпке.
9. Трансляція модальної логіки в логіку предикатів.

Контрольні запитання до частини 2.

1. Лінійна пропозиційна логіка. Специфікації у ЛПТЛ. Особливості пропозиційного рівня. Зв'язок з теорією автоматів.
2. Транзиційні системи. Основні типи, об'єкти ТС. Шляхи, слова дій та логічних значень.
3. Недетермінізм ТС. Логіки розгалуженого часу. Специфікації властивостей ТС.
4. Навести приклади моделей реальних систем та специфікації їх властивостей.
5. Багатозначні логіки. Тризначна логіка Лукасевича. Теорема дедукції в цій логіці.
6. Алгоритм перевірки виконуваності формул логіки Лукасевича.
7. Композиції ТС та мережі Петрі.

Теми для самостійної роботи № 1.

1. Методи перевірки виконуваності формул класичних логік на моделях.
2. Метод резолюцій та семантичного табло для цих логік.
3. Принципова різниця цих методів для логіки висловлювань та логіки предикатів.
4. Проблема уніфікації та її розв'язання для теорій першого порядку.
5. АК-уніфікація та системи діофантових рівнянь.

6. Складність алгоритмів. Навести приклади реальних систем та їх специфікацій в цих логіках, з перевіркою їх виконуваності на цих системах.
7. Теми для самостійної роботи № 2.

Теми для самостійної роботи № 2.

1. Знайомство з модальними, темпоральними та дескриптивними логіками.
2. Методи перевірки виконуваності специфікацій в цих логіках.
3. Зв'язок дескриптивних логік і баз знань.
4. Мова запитів до бази знань та алгоритми генерації відповідей на запити.
5. Динамічна логіка та логіка Хоара.
6. Програмні інваріанти, їх пошук та застосування в процесі верифікації програмного забезпечення.

9. Рекомендовані джерела.

Основні:

1. Papadimitriou C.R. Computational complexity. – Addison-Wesley – 1994. – 532 p.
2. Fisher M.J., Ladner R.E. Propositional Modal Logic of Programs. In Proc. 9-th ACM Ann. Symposium on Theory of Computing. – 1977. – P. 286-294.
3. Fisher M.J., Ladner R.E. Propositional Dynamic Logic of regular programs. J. Comp. System Sci. – 1979. – v.18. – N 2. – P. 194-211.
4. Goldblatt R. Logics of Time and Computation. – Lecture Notes. – N 7. Center for the Study of Language and Information. – 1987. – P. 1-27.
5. Ben-Ari M. Mathematical Logic for Computer Science. – Prentice Hall International (UK) Ltd. – 1993. – 305 p.
6. Bryant R.E. Symbolic Boolean Manipulation with Ordered Binary Decision Diagrams. School of Comp. Science, Carnegie Mellon University, Pittsburg. – 1992 (June). – 34 p.
7. Burch J.R., Clarke E.M., McMillan K.I. Symbolic model checking. - Fifth Annual IEEE Symposium on Logic in Computer Science (Philadelphia, June). IEEE. – New York. – 1990. – P. 428-439.
9. Burch J.R., Clarke E.M., Dill D.L., McMillan K.I. Sequential circuit verification using symbolic model checking. – Proceed. of the 27-th ACM/IEEE Design Automation Conference (Orlando, June). ACM. – New York. – 1990. – P. 46-51.
10. Clarke E.M., Emerson E.A., Sistla A.P. Automatic verification of finite-state concurrent systems Using temporal logic specifications. – ACM Transactions on Programming Languages. – N 2 (April). – 1986. – P. 214-263.
11. Clarke E.M., Grumberg Jr. O., Peled D. Model Checking. – The MIT Press: Cambridge, Massachusetts, London, England. – 2001. – 356 p.

12. Emerson E.A. Temporal and modal logics. – Handbook of Theoretical Computer Science: Elsevier. – v. B. – 1990. – P. 995-1072.
13. Baader F., Calvanese D., McGuinness D.L. and other. The Description Logic Handbook. Cambridge: University Press. – 2007. – 601 p.

Додаткові:

1. Gabbay D. Elementary Logic (A procedural perspective). – Prentice Hall Europe. – 1998. – 359 p.
2. Лисовик Л.П., Редько В.Н. Алгоритмы и формальные системы. – К., 1981.
3. Arnold A. Finite Transition Systems: Semantics of Communicating Systems. – Paris: Prentice Hall. – 1994. – 177 p.
4. Emerson E. A., Sistla A.P. Deciding full branching time logic. Information and Control. – v. 61. – 1984. – P. 175-201.