

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Кафедра теорії та технології програмування

«ЗАТВЕРДЖУЮ»

Заступник декана
з навчальної роботи

_____ Кашпур О.Ф.

« ____ » _____ 2019 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ПРОГРАМНІ ЛОГІКИ ТА ЇХ ЗАСТОСУВАННЯ**

для студентів

галузь знань	12 «Інформаційні технології» <i>(шифр і назва)</i>
спеціальність	122 «Комп'ютерні науки» <i>(шифр і назва спеціальності)</i>
освітній рівень	магістр <i>(молодший бакалавр, бакалавр, магістр)</i>
освітня програма	«Інформатика» <i>(назва освітньої програми)</i>
спеціалізація	«Теорія та технологія програмування» <i>(назва спеціалізації)</i>
вид дисципліни	вибіркова

Форма навчання	денна
Навчальний рік	2019/2020
Семестр	3
Кількість кредитів ECTS	5
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	залік

Викладачі: **асистент Шишацька О.В.**

Пролонговано: на 20__/20__ н.р. _____ (_____) «__» 20__ р.
(підпис, ПІБ, дата)

на 20__/20__ н.р. _____ (_____) «__» 20__ р.
(підпис, ПІБ, дата)

КИЇВ – 2019

Розробник: Шишацька Олена Володимирівна, асистент кафедри «Теорії та технології програмування»

ЗАТВЕРДЖЕНО

Зав. кафедри «Теорії та технології програмування»

(підпис)

Нікітченко М.С.
(прізвище та ініціали)

Протокол № ____ від « ____ » _____ 20__ р.

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

Протокол від « ____ » _____ 20__ року № ____

Голова науково-методичної комісії _____
(підпис)

Омельчук Л.Л.
(прізвище та ініціали)

« ____ » _____ 20__ року

Затверджено вченою радою факультету комп'ютерних наук та кібернетики

Протокол від « ____ » _____ 20__ року № ____

Голова Вченої ради _____
(підпис)

Анісімов А.В.
(прізвище та ініціали)

1. Мета дисципліни – поглиблення знань з теорії програмування та засвоєння знань з програмних логік. Розуміння принципів побудови програмних логік, їх застосування для доведення коректності програмних систем, застосування формальних методів при розробці інформаційних систем.

2. Попередні вимоги до опанування або вибору навчальної дисципліни (за наявності):

1. *Знати:* основні поняття програмування, методи формалізації мов програмування та мов специфікацій, методи моделювання предметних областей; логічні числення.
2. *Вміти:* формалізувати мови специфікацій та програм, моделювати предметні області за допомогою відповідних мов, застосувати програмні засоби аналізу специфікацій.
3. *Володіти елементарними навичками:* програмування в сучасних мовах, перевірки виконаності формул.

3. Анотація навчальної дисципліни:

Навчальна дисципліна “Програмні логіки та їх застосування” є складовою освітньо-наукової програми підготовки фахівців за освітньо-кваліфікаційним рівнем «магістр» галузі знань 12 „Інформаційні технології” зі спеціальності 122 „Комп’ютерні науки”, спеціалізації „Теорія та технологія програмування” освітньо-наукової програми „Інформатика”.

Дана дисципліна є навчальною дисципліною за вибором за *програмою “Інформатика” за спеціалізацією “Теорія та технологія програмування”*.

Викладається у 3 семестрі 2 курсу магістратури в обсязі – 150 год.

(5 кредитів ECTS) зокрема: семінарські заняття – 44 год., самостійна робота – 106 год. У курсі передбачено 3 частини та 3 контрольні роботи. Завершується дисципліна – заліком у 3 семестрі.

В результаті вивчення навчальної дисципліни студент повинен:

знати основні поняття і методи теорії програмування, математичної логіки, ієрархію логік, логічні числення, формальні методи.

вміти формалізувати специфікації програмного забезпечення за допомогою апарату програмних логік, аналізувати істинність побудованих специфікацій, будувати виведення в відповідних логічних численнях.

Для допуску до дисципліни „Програмні логіки та їх застосування” освітньо-професійної програми «Інформатика» студент повинен опанувати компетентності та результати навчання, які надає дисципліна „Формальні методи розробки програмних систем” програми «Інформатика».

4. Завдання (навчальні цілі):

набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень у програмуванні, відповідно до освітньої кваліфікації “Магістр з комп’ютерних наук”. Зокрема, розвивати:

- розуміння економічних преференцій інноваційного розвитку ІТ підприємств (новітні підходи організації, застосування програмних, апаратних, мережних, математичних, технологічних, ергономічних та інших засобів) з метою вирішення актуальних задач підвищення конкурентоспроможності галузі; здатність розв’язувати складні задачі і проблеми проектування корпоративного інформаційного середовища, що передбачає здійснення інновацій;

- здатність до алгоритмічного та логічного мислення.

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 методи теорії програмування, математичної логіки	Семінарські заняття	Контрольна робота 1, 60% правильних відповідей	15%
РН1.2	Знати ієрархію логік, логічні числення, формальні методи	Семінарські заняття	Контрольна робота 2, 3, 60% правильних відповідей	20%
РН2.1	Вміти формалізувати специфікації програмного забезпечення за допомогою апарату програмних логік, аналізувати істинність побудованих специфікацій	Семінарські заняття	Контрольна робота 1, 60% правильних відповідей	20%
РН2.2	Вміти будувати виведення в відповідних логічних численнях	Семінарські заняття	Контрольна робота 2, 3, 60% правильних відповідей	25%
РН3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань використання формальних специфікацій	Самостійна робота	Поточне оцінювання	10%
РН4.1	Організувати свою самостійну роботу для досягнення результату	Самостійна робота	Поточне оцінювання	5%
РН4.2	Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість	Самостійна робота	Поточне оцінювання	5%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	РН 1.1	РН 1.2	РН 2.1	РН 2.2	РН 3.1	РН 4.1	РН 4.2
Програмні результати навчання (з опису освітньої програми)							
ВПРН2.3. Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.	+		+		+	+	
ВПРН2.5. Знати та вміти застосовувати програмно-орієнтовані логічні формалізми.		+		+	+		+

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

- семестрове оцінювання:

1. Контрольна робота 1: РН 1.1, РН 2.1 — 35 балів/21 балів.

2. Контрольна робота 2: РН1.2, РН2.2 - 20 балів/12 балів.

3. Контрольна робота 3: РН1.2, РН2.2 - 25 балів/15 балів.

- підсумкове оцінювання у формі заліку. Виставляється за результатами роботи студентами впродовж усього семестру та не передбачає додаткових заходів оцінювання для успішних студентів.

7.2 Організація оцінювання:

Терміни проведення форм оцінювання:

1. Контрольна робота 1: до 5 тижня семестру.

2. Контрольна робота 2: до 9 тижня семестру.

3. Контрольна робота 3: до 13 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.

7.3 Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59
Зараховано / Passed	60-100
Не зараховано / Fail	0-59

8. Структура навчальної дисципліни. Тематичний план лекцій і лабораторних занять

№ лекції	Назва лекції	Кількість годин		
		Семін. зан.	Практ. зан.	Сам. р-та
Частина 1. Основні підходи до верифікації та подання семантики програмного забезпечення				
1.	Тема 1. Основні поняття. Системи автоматичного доведення теорем.	2		6
2.	Тема 2. Перевірка моделей.	2		6
3.	Тема 3. Семантика програм. Алгебраїчна та аксіоматична семантика.	2		6
4.	Тема 4. Семантика програм. Операційна семантика.	2		6
5.	Тема 5. Семантика програм. Денотаційна семантика.	2		6
6.	Тема 6. Семантика перетворювачів предикатів.	4		7
	Контрольна робота 1			
Всього по частині 1		14		37
Частина 2. Програмні логіки				
7.	Тема 7. Динамічна логіка.	4		6
8.	Тема 8. Пропозиційна динамічна логіка, аксіоматизація динамічної логіки.	2		6
9.	Тема 9. Динамічна логіка першого порядку.	2		6
10.	Тема 10. Логіка розділення та логіка згрупованих імплікацій.	4		6
11.	Тема 11. Логіка співставлення.	2		7
	Контрольна робота 2			
Всього по частині 2		14		31
Частина 3. Формальні методи для паралельних програмних систем				
12.	Тема 12. Застосування логіки Флойда-Хоара для паралельних програмних систем. Метод Гріс-Овікі.	2		6
13.	Тема 13. Паралельна динамічна логіка.	4		6
14.	Тема 14. Мю-числення.	4		6
15.	Тема 15. Взаємодіючі послідовні процеси.	2		6
16.	Тема 16. Числення взаємодіючих систем.	2		6
17.	Тема 17. Пі-числення.	2		8
	Контрольна робота 3			
Всього по частині 3		16		38
ВСЬОГО		44		106

Загальний обсяг **150 год.**, в тому числі:

Семінарських занять – **44 год.**.

Самостійна робота - **106 год.**

Теми, винесені на самостійне вивчення:

Розвиток систем автоматичного доведення теорем. Формули для обчислення найсильнішої післяумови та сильного інваріанту. Синтаксис динамічної логіки першого порядку. Властивості динамічної логіки першого порядку. Порівняння логіки співставлення та логіки розділення. Властивості логіки співставлення.

9. Рекомендовані джерела:

Основна

1. Нікітченко М.С., Шкільняк С.С. Прикладна логіка. – К., 2013.
2. Schneider K.: Verification of Reactive Systems. Formal Methods and Algorithms. – Berlin-Heidelberg: Springer-Verlag, 2004.
3. Nielson H.R. Semantics with Applications: A Formal Introduction / H.R. Nielson, F. Nielson // John Wiley & Sons Inc., 1992 P. 240.
4. Dijkstra E.W. A Discipline of Programming / E.W. Dijkstra // Prentice-Hall, Englewood Cliffs, New Jersey, 1976.

Додаткова:

5. Lamport L. Win and sin: Predicate Transformers for Concurrency / L. Lamport // ACM Transactions on Programming Languages and Systems Vol. 12, 1990 P. 396-428.
6. Harel D. Dynamic logic, Handbook of Philosophical Logic / D. Harel, D. Kozen, J. Tiuryn // 1984 P. 497-604.
7. O’Hearn P.W. The logic of bunched implications / P.W. O’Hearn, D. J. Pym // Bulletin of Symbolic Logic, Vol. 5(2), 1999 P. 215–244.
8. Reynolds J.C. Separation Logic: A logic for Shared Mutable Data Structures / J.C. Reynolds // LICS, 2002 P. 55-74.
9. Rosu G., Ellison C., Schulte W. Matching Logic: An Alternative to Hoare/Floyd Logic LNCS 6486, 2010 P. 142-162.
10. Owicki S.S. Verifying properties of parallel programs: An axiomatic approach / S.S. Owicki, D. Gries // Communications of the ACM, Vol. 19(5), 1976 P. 279–285.
11. Bergstra J.A. et al Handbokk of Process Algebra – Holland 2001.
12. Hoare C. A. R. Communicating sequential processes. *Communications of the ACM* **21** (8): P. 666–677 , 1978.
13. Milner R. A Calculus of Communicating Systems, Springer Verlag, 1980.