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.

«____»_____2018 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ФОРМАЛЬНІ МЕТОДИ РОЗРОБКИ ПРОГРАМНИХ СИСТЕМ/ FORMAL METHODS IN SOFTWARE DEVELOPMENT

для студентів

галузь знань **12 «Інформаційні технології»**
(шифр і назва)

спеціальність **122 «Комп'ютерні науки»/ 122 «Computer Science»**
(шифр і назва спеціальності)

освітній рівень **магістр**
(молодший бакалавр, бакалавр, магістр)

освітня програма **«Штучний інтелект»/ "Artificial Intelligence"**
(назва освітньої програми)

вид дисципліни обов'язкова

Форма навчання	**денна**
Навчальний рік	**2018/2019**
Семестр	**2**
Кількість кредитів ECTS	**5**
Мова викладання, навчання та оцінювання	**англійська, українська/ Ukrainian, English**
Форма заключного контролю	**екзамен**

Викладачі: **д.ф.-м.н., проф. Нікітченко М.С.** (лекції)

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

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

КИЇВ – 2018

Розробник: Нікітченко Микола Степанович, д.ф.-м.н., професор кафедри «Теорії та технології програмування»

ЗАТВЕРДЖЕНО
В.о. зав. кафедри «Теорії та технології програмування»

_____ Панченко Т.В.
(підпис) (прізвище та ініціали)

Протокол № ___ від «____» _____ 2018 р.

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

Протокол від «____» _____ 2018 року №___

Голова науково-методичної комісії _____ Хусаїнов Д.Я.
(підпис) (прізвище та ініціали)

Затверджено вченою радою факультету комп'ютерних наук та кібернетики

Протокол від «____» _____ 2018 року №___

Голова вченої ради факультету _____ А.В. Анісімов

© Нікітченко М.С., 2018 рік

ВСТУП

1. Мета дисципліни – засвоєння основних концепцій, принципів та понять сучасних методів розробки програмних систем та їх застосування для адекватного моделювання мов специфікацій і програмування та використання побудованих моделей для створення сучасних програмних та інформаційних систем високої якості.

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

1. Знати: основні поняття, засоби і методи математичної логіки, їх застосування в інформатиці й програмуванні; знати мови пропозиційної логіки та логіки 1-го порядку, їх можливості для опису предметних областей; основні методи пошуку доведень та засоби логічного виведення.

2. Вміти: описувати на формальних мовах 1-го порядку твердження стосовно тих чи інших предметних областей; встановлювати істинність пропозиційних формул, безквантворних формул, формул 1-го порядку; встановлювати наявність логічного наслідку; встановлювати виразність та невиразність предикатів у моделях мови.

3. Володіти елементарними навичками: програмування в сучасних мовах, перевірки виконуваності формул.

3. Анотація навчальної дисципліни:

Навчальна дисципліна «Формальні методи розробки програмних систем» є складовою освітньо-наукової програми підготовки спеціалістів за освітньо-кваліфікаційним рівнем «магістр» галузі 12 „Інформаційні технології” зі *спеціальності 122* „Комп’ютерні науки”, *освітньо-наукової програми* **–** „Штучний інтелект”.

Дана дисципліна є обов’язковою навчальною дисципліною за ***програмою “Штучний інтелект”***.

Викладається в 2 семестрі 1 курсу магістратури в обсязі 150 годин.

(5 кредитів ECTS)) зокрема: *лекції – 36 год., консультації – 2 год., самостійна робота – 112 год.* У курсі передбачено **2** *частини* та 2 *контрольні роботи*. Завершується дисципліна – **екзаменом в 2 семестрі.**

В результаті вивчення навчальної дисципліни студент повинен:

знати: основні поняття програмування, методи формалізації мов програмування та мов специфікацій, методи моделювання предметних областей; логічні числення.

вміти: формалізувати мови специфікацій та програм, моделювати предметні області за допомогою відповідних мов, застосувати програмні засоби аналізу специфікацій.

4. Завдання (навчальні цілі):

набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень у програмуванні, відповідно освітньої кваліфікації «Магістр з комп’ютерних наук».

Зокрема, розвивати:

- здатність до абстрактного мислення, аналізу та синтезу;
- здатність спілкуватися іноземною мовою.

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 програмування та логічні числення.*	*Лекція*	*Контрольна робота 60% правильних відповідей, екзамен*	20%
РН1.2	*Знати методи формалізації мов програмування та мов специфікацій.*	*Лекція*	*Контрольна робота 60% правильних відповідей, екзамен*	20%
РН1.3	*Знати методи моделювання предметних областей.*	*Лекція*	*Контрольна робота 60% правильних відповідей, екзамен*	20%
РН2.1	*Вміти формалізувати мови специфікацій та програм, моделювати предметні області за допомогою відповідних мов, застосувати програмні засоби аналізу специфікацій.*	*Лекція, самостійна робота*	*Поточне оцінювання, екзамен*	20%
РН3.1	*Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування та розробки специфікацій програм.*	*Лекція*	*Поточне оцінювання, екзамен*	10%
РН4.1	*Організовувати свою самостійну роботу для досягнення результату.*	*Екзамен, самостійна робота*	*Поточне оцінювання, екзамен*	10%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни **Програмні результати навчання**	**РН 1.1**	**РН 1.2**	**РН 1.3**	**РН 2.1**	**РН 3.1**	**РН 4.1**
(з опису освітньої програми)						
ПР4. Аналізувати великі дані та моделювати високорівневі абстракції у великих наборах даних різної природи, проектувати сховища великих даних, для видобутку даних і знань, візуалізовувати великі дані, будувати і оцінювати регресивні моделі, що генеруються на основі великих даних		+	+			
ПР13. Використовувати знання з комп'ютерних наук та інформаційних технологій й уміння критичного мислення, аналізу та синтезу в професійних цілях.	+					+
ПР15. Володіти методами розробки та впровадження заходів, спрямованих на підвищення ефективності інформаційних систем.				+	+	

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

- семестрове оцінювання:

1. Контрольна робота 1: РН 1.1., РН 1.2 — 30 балів/18 балів.

2. Контрольна робота 2: РН1.3 - 30 балів/18 балів.

- підсумкове оцінювання (у формі екзамену)*:*

- максимальна кількість балів які можуть бути отримані студентом: 40 балів;

- результати навчання які будуть оцінюватись: РН1.1, РН1.2, РН1.3, РН2.1, РН3.1, РН4.1;

- форма проведення і види завдань: письмова.

Види завдань: 4 письмових питання.

- 1 питання: *РН1.1, РН3.1, РН4.1;*
- 2 питання: *РН1.2, РН3.1, РН4.1;*
- 3 питання: *РН1.3, РН3.1, РН4.1;*
- 4 питання: *РН2.1, РН3.1, РН4.1;*

За розгорнуту відповідь на кожне завдання студент може отримати від 1 до 10 балів.

Критерії оцінювання відповіді студента на питання:

- повнота розкриття питання 1-4 бали;
- логіка викладення 2 бал;
- аналітичні міркування 1-4 бали.

Типове завдання контрольної роботи 1:

Теоретичні питання до контрольної роботи 1:

1. Властивості основної пентади програмування.
2. Властивості програмної пентади.
3. Що таке часткова коректність програм? Яким чином доводиться часткова коректність програм?
4. Що таке повна коректність програм? Яким чином доводиться повна коректність програм?
5. Розкрийте зміст формалізації поняття програми.
6. Визначте різні класи функцій.
7. Визначте програмні системи різного рівня абстракції.
8. Дайте визначення класу номінативних даних.
9. Повний клас обчислюваних функцій над номінативними даними.

Завдання для самостійної роботи. Побудувати моделі та специфікації наступних програмних систем:

1. Банкомат
2. Кавовий автомат
3. Склад
4. Е-магазин
5. Мікрохвильова піч
6. Телефон
7. Електроплита

Контрольні запитання до частини 1:

1. Які принципи є основними методологічними принципами теорії програмування?
2. Як формулюються принципи розвитку та гносеологічності?
3. Як визначається пентода основних понять програмування?

4. Як визначаються поняття користувача, проблеми, програми, обчислюваності, програмування?
5. Які властивості основних понять програмування?
6. Як аспекти програм є головними?
7. На підставі яких принципів відбувається формалізація програмних понять?
8. Як визначаються інтенсіональні та екстенсіональні аспекти програмних понять?
9. Як визначаються системи різних рівнів абстракції?
10. Як визначаються мови специфікацій та програмування?
11. Які є формальні моделі обчислюваних функцій?
12. Як визначається обчислюваність над складними структурами даних?
13. Як визначається обчислюваність над номінативними даними?
14. Що таке натуральна обчислюваність?
15. Як визначається обчислюваність композицій програм?
16. Повні класи обчислюваних функцій.
17. Якими методами описують предметні області?
18. Як методи використовують для специфікації вимог до програмних систем?
19. Які засади RAISE-методу розробки програм?
20. Які логічні формалізми використовують для специфікацій програм?
21. Як використовується класична та некласична логіка для специфікацій програм?
22. Як визначаються темпоральні та модальні логіки?
23. Як визначаються аксіоматичні методи специфікації програм?
24. Як визначається логіка Флойда-Хоара та які властивості вона має?
25. Які особливості має семантико-синтаксична технологія розробки програм?
26. Які особливості має метод послідовних уточнень?
27. Які особливості у розробку програм вносять об'єктно-орієнтовані методи?
28. Яка мета стандартів програмування?
29. Як використовують технологічні та інструментальні засоби специфікації та розробки програм?

***Рекомендована література**: [1–5].*

Типове завдання контрольної роботи 2:

Верифікувати з використання програмних засобів моделі наступних програмних систем:
1. Банкомат
2. Кавовий автомат
3. Склад
4. Е-магазин
5. Мікрохвильова піч
6. Телефон
7. Електроплита

Контрольні запитання до частини 2:

1. Навести приклади станів транзиційних систем.
2. Визначити логіки Флойда-Хоара. Сформулювати їх аксіоматику.
3. Довести коректність числення та відносну повноту логіки Флойда-Хоара.
4. Засоби специфікації та розробки програм за допомогою логік Флойда-Хоара.
5. Основні засади методу TLA та побудувати приклади.
6. Сформулювати властивості досяжності і безпеки.
7. Надати визначення транзиційного переходу.
8. Принципи верифікації в TLA.
9. Принципи перевірки моделей, заданих в TLA, за допомогою SPIN.
10. Принципи перевірки моделей, заданих в Z.
11. Властивості Z-методу.

12. Принципи перевірки моделей, заданих в B.
13. Властивості B-методу.
14. Принципи перевірки моделей, заданих в RAISE.
15. Властивості RAISE-методу.

***Рекомендована література**: [1–5, 7–13].*

Запитання для підготовки до екзамену

1. Які принципи є основними методологічними принципами теорії програмування?
2. Як формулюються принципи розвитку та гносеологічності?
3. Як визначається пентада основних понять програмування?
4. Як визначаються поняття користувача, проблеми, програми, обчислюваності, програмування?
5. Які властивості основних понять програмування?
6. Властивості основної пентади програмування.
7. Властивості програмної пентади.
8. Що таке часткова коректність програм? Яким чином доводиться часткова коректність програм?
9. Що таке повна коректність програм? Яким чином доводиться повна коректність програм?
10. Розкрийте зміст формалізації поняття програми.
11. Визначте різні класи функцій.
12. Визначте програмні системи різного рівня абстракції.
13. Дайте визначення класу номінативних даних.
14. Повний клас обчислюваних функцій над номінативними даними.
15. Які аспекти програм є головними?
16. На підставі яких принципів відбувається формалізація програмних понять?
17. Класи функцій, що використовуються для формалізації програм.
18. Як визначаються інтенсіональні та екстенсіональні аспекти програмних понять?
19. Поняття композиційно-номінативної системи.
20. Як визначаються системи різних рівнів абстракції?
21. Як визначаються мови специфікацій та програмування?
22. Якими методами описують предметні області?
23. Як методи використовують для специфікації вимог до програмних систем?
24. Які засади RAISE-методу розробки програм?
25. Які засади B-методу розробки програм?
26. Які засади Z-методу розробки програм?
27. Які засади TLA-методу розробки програм?
28. Які логічні формалізми використовують для специфікацій програм?
29. Як використовується класична та некласична логіка для специфікацій програм?
30. Як визначаються темпоральні та модальні логіки?
31. Як визначаються аксіоматичні методи специфікації програм?
32. Як визначається логіка Флойда-Хоара та які властивості вона має?
33. Повнота логіки Флойда-Хоара.
34. Які особливості має семантико-синтаксична технологія розробки програм?
35. Які особливості має метод послідовних уточнень?
36. Які особливості у розробку програм вносять об’єктно-орієнтовані методи?
37. Яка мета стандартів програмування?
38. Як використовують технологічні та інструментальні засоби специфікації та розробки програм?

Студент не допускається до екзамену, якщо під час семестру набрав менше ніж 24 балів.

7.2. Організація оцінювання:

Терміни проведення форм оцінювання:

1. Контрольна робота 1: до 5 тижня семестру.

2. Контрольна робота 2: до 12 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

№ лекції	Назва лекції	Кількість годин		
		Лекції	Практ занять	Сам. р-та
	Частина 1. Формальні моделі програм та методи їх специфікації			
1	**Тема 1.** Вступ до предмету. Основні методологічні принципи побудови формальних моделей програм.	2		6
2	**Тема 2**. Принципи формалізації програмних понять. Інтенсіонал та екстенсіонал понять множини, функції та програми.	2		6
3	**Тема 3.** Мови специфікації та програмування	2		6
4	**Тема 4.** Формальні моделі обчислюваних функцій	2		6
5	**Тема 5.** Предметні області та методи їх опису. Методи розробки програм.	2		6
6	**Тема 6.** Структури даних та класи функцій у мовах специфікацій	2		6
7	**Тема 7.** Класи композицій у мовах специфікацій	2		6
8	**Тема 8.** Методи уточнення даних, функцій та композицій.	2		6
9	**Тема 9.** Приклади специфікацій програмних систем	2		6
Контрольна робота 1				
Всього по частині 1		18	1	54
	Частина 2. Методи верифікації програм			
10.	**Тема 10.** Верифікація в логіка Флойда-Хоара.	2		6
11.	**Тема 11.** Приклади застосування. Коректність та повнота логіки.	2		6
12.	**Тема 12.** Технологічні та інструментальні засоби специфікації та розробки програм за допомогою логік Флойда-Хоара.	2		6
13.	**Тема 13.** Верифікація систем в TLA. Загальні положення.	2		6
14.	**Тема 14.** Верифікація систем в TLA. Приклади.	2		8
15.	**Тема 15.** Перевірка моделей за допомогою SPIN.	2		6
16.	**Тема 16.** Верифікація систем в Z.	2		6
17.	**Тема 17.** Верифікація систем в B.	2		8
18.	**Тема 18.** Верифікація систем в RAISE.	2		6
Контрольна робота 2				
Всього по частині 2		18		58
	Консультація		2	
	Екзамен			
ВСЬОГО		36	2	112

Загальний обсяг ***150*** *год.,* в тому числі:
Лекцій **–** ***36*** *год.*
Консультації – ***2*** *год.*
Самостійна робота **-** ***112*** *год.*

Теми, винесені на самостійне вивчення:

Основні поняття програмування. Формальні моделі програм. Формалізація основних понять. Методи формалізації мов програмування та специфікацій. Приклади специфікацій простих систем [1,4,5].

Рівні абстракції в розгляді основних понять програмування. Властивості основних понять програмування. Головні аспекти програм. Формалізми подання синтаксис та семантика програм [1,2,4,5].

Теоретико-функціональна та теоретико-номінатна платформи формалізації програмних понять. Інтенсіонал поняття, екстенсіонал поняття. Класи не детермінованих функцій. Формалізація композицій [1,2,4,5].

Сформулювати відмінності між теоретико-функціональною та теоретико-номінативною платформою формалізації програмних понять. Навести приклади. Надати визначення інтенсіоналу поняття та екстенсіоналу. Навести приклади. Визначити класи не детермінованих функцій. Розглянути методи формалізація композицій [1, 2, 4, 5].

Сформулювати особливості мов специфікацій у порівнянні з мовами програмування. Визначити денотативні композиції в мовах специфікацій. Які логічні композиції використовують у мовах специфікацій [1–5]?

Традиційні формальні моделі обчислюваних функцій та їх обмеженість. Моделі абстрактної обчислюваності. Аксіоматичні визначення обчислюваних функцій [1,4,5].

Розглянути традиційні формальні моделі обчислюваних функцій та їх сформулювати випадки їх обмеженості. Перерахувати основні моделі абстрактної обчислюваності. Розглянути аксіоматичні визначення обчислюваних функцій. [1, 4, 5].

Номінативні дані та їх класифікація. Базові функції. Моделювання натуральних чисел номінативними даними. Композиції номінативних функцій [1, 4, 5].

Дати визначення даним скінченої структури. Сформулювати визначення функцій натуралізації та денатуралізації. Розглянути схему натуральної обчислюваності. Довести теореми про повні класи обчислюваних функцій [1,4,5].

Розглянути уточнення даних на підставі схеми Хоара. Навести приклади. Довести властивості уточнених даних [1, 6].

Класифікувати класи функцій у мовах специфікацій. Навести приклади використання однозначних та багатозначних функцій. Розглянути способи подання та перетворення функцій [1–3].

Класифікувати класи композицій у мовах специфікацій. Визначити коннотативні та денотативні композиції. Сформулювати їх властивості. Визначити основні методи подання та перетворення композицій [1–3].

Побудувати моделі та специфікації обраних програмних систем.

9. Рекомендовані джерела:

Основна

1. М.С. Нікітченко, Теорія програмування: Частина 1.– Ніжин: Видавництво НДУ імені Миколи Гоголя, 2010.– 119 с.
2. М.С. Нікітченко, С.С. Шкільняк. Математична логіка та теорія алгоритмів. – К., 2008.
3. И.А. Басараб, Н.С.Никитченко, В.Н. Редько. Композиционные базы данных. - К., Либідь, 1992.– 182 с.
4. The RAISE specification language. Prentice Hall Int.– 1992.– 397 p.
5. С. Лавров. Программирование. Математические основы, средства, теория.– СПб.: БХВ-Петербург, 2001.– 320 с.
6. Бабенко Л.П., Лавріщева К.М. Основи програмної інженерії: Навч. посіб.–К.: Т-во "Знання", 2001.– 269 с.

Додаткова

7. Hoare C.A.R., Jifeng He. Unifying Theories of Programming.– London: Prentice Hall Europe, 1998.– 298 p.
8. Schneider K.: Verification of Reactive Systems. Formal Methods and Algorithms. Springer-Verlag Berlin Heidelberg (2004)
9. Clarke E.M., Grumberg O., Peled D.: Model Checking. MIT Press (1999)
10. Mike Spivey. The Z Notation: A Reference Manual, 2nd edition. Prentice Hall International Series in Computer Science, 1992.
11. Jim Davies and Jim Woodcock. Using Z: Specification, Refinement and Proof. Prentice Hall International Series in Computer Science, 1996.
12. Jean-Raymond Abrial. Assigning Programs to Meanings, Cambridge University Press, 1996. ISBN 0-521-49619-5.
13. Steve Schneider. The B-Method: An Introduction, Cornerstones of Computing series, 2001. ISBN 0-333-79284-X.
14. Leslie Lamport. Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers, 2002 Pearson Education Publ.