



НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ
Факультет інформаційних технологій
Кафедра комп'ютерних наук


«ЗАТВЕРДЖУЮ»
Проректор з науково-педагогічної
роботи та розвитку
М. Кваша
«» 20 21 р.



РОЗГЛЯНУТО І СХВАЛЕНО

На засіданні вченої ради факультету
інформаційних технологій
Протокол № 2 від "23" 09 2021 р.
Декан  ФАКУЛЬТЕТ
ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ, Глазунова О.Г.



На засіданні кафедри комп'ютерних наук
Протокол № 1 від "20" 09 2021 р.
зав. кафедри
 к.т.н., доц. Голуб Б. Л.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

“ФОРМАЛЬНІ МЕТОДИ РОЗРОБКИ ПРОГРАМНИХ СИСТЕМ”

Галузь знань	12 інформаційні технології
Спеціальність	122 Комп'ютерні науки
Рівень вищої освіти	Третій (освітньо-науковий) рівень
Розробник:	к.т.н., доц. Ткаченко О.М.

Київ – 2021

1. Опис навчальної дисципліни
ФОРМАЛЬНІ МЕТОДИ РОЗРОБКИ ПРОГРАМНИХ СИСТЕМ

Галузь знань, напрям підготовки, спеціальність, освітньо-кваліфікаційний рівень		
Галузь знань	12 «Інформаційні технології»	
Освітньо-науковий рівень	Третій	
Освітній ступінь	Доктор філософії	
Спеціальність	122 комп'ютерні науки	
Освітньо-наукова програма	Інформаційні технології	
Характеристика навчальної дисципліни		
Вид	Дисципліна циклу спеціальної (фахової) підготовки Нормативна	
Загальна кількість годин	90	
Кількість кредитів ECTS	3	
Кількість змістових модулів	2	
Курсовий проект (робота) (якщо є в робочому навчальному плані)	-	
Форма контролю	Іспит	
Показники навчальної дисципліни для денної та заочної форм навчання		
	денна форма навчання	заочна форма навчання
Рік підготовки	1	
Семестр	2	
Лекційні заняття	10 год.	
Семінарські заняття	20 год.	
Лабораторні заняття		
Самостійна робота	60 год	
Індивідуальні завдання	0 год.	
Кількість тижневих годин для денної форми навчання: аудиторних самостійної роботи	4 год. 5 год.	

2. Мета та завдання навчальної дисципліни

Мета: поглиблення знань щодо концепцій, принципів та понять формальних методів розробки програмних систем та їх застосування для створення якісних і надійних програмних систем.

Завдання навчальної дисципліни:

У результаті вивчення навчальної дисципліни аспірант повинен

Знати:

- логічні числення;
- поглиблені методи формалізації мов програмування;
- методи мов специфікацій;
- поглиблені методи моделювання предметних областей;
- поглиблені методи специфікації програмних систем;
- поглиблені методи верифікації програмних систем.

Вміти:

- використовувати засоби формальної логіки для опису предметних областей;
- встановлювати істинності пропозиційних формул;
- встановлювати наявність логічного наслідку;
- встановлювати виразність предикатів у моделях мови;
- використовувати засоби формальної специфікації програмних систем;
- використовувати формальні методи верифікації програмних систем.

Опанування дисципліни покликане сприяти досягненню відповідних результатів навчання (РН), а також загальних (ЗК) і спеціальних (фахових, СК) компетентностей:

ЗК01. Здатність до абстрактного мислення, аналізу та синтезу

ЗК04. Здатність розробляти проєкти та управляти ними.

СК03. Здатність застосовувати сучасні інформаційні технології, бази даних та інші електронні ресурси, спеціалізоване програмне забезпечення у науковій та навчальній діяльності.

СК06. Здатність ініціювати, розробляти і реалізовувати комплексні інноваційні проєкти у галузі комп'ютерних наук та дотичні до неї міждисциплінарні проєкти, лідерство під час їх реалізації.

СК09 Здатність до продукування нових ідей і розв'язання комплексних проблем у галузі комп'ютерних наук, а також до застосування сучасних методологій, методів та інструментів педагогічної та наукової діяльності в комп'ютерних науках.

РН01. 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.

PH04. 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'ютерній науці та дотичних міждисциплінарних напрямках.

PH07. 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.

PH10. Здійснювати пошук та критичний аналіз інформації, концептуалізацію та реалізацію наукових проектів з комп'ютерних наук.

3. Програма та структура навчальної дисципліни

ЗМІСТОВНИЙ МОДУЛЬ 1. ПРИНЦИПИ ФОРМАЛЬНОГО ПІДХОДУ ДО РОЗРОБКИ ПРОГРАМ ТА СПЕЦИФІКАЦІЯ

ТЕМА 1. МОВИ СПЕЦИФІКАЦІЙ. ФОРМАЛЬНІ МОДЕЛІ ОБЧИСЛЮВАЛЬНИХ ФУНКЦІЙ

Принципи формалізації програмних понять. Визначення мови специфікації програм. Приклади та можливості мов специфікацій. Класи мов програмування. Парадигми програмування.

Формальні моделі обчислюваних функцій. Обчислюваність над складними структурами даних. Номінативні дані та обчислюваність над ними. Композиційність та обчислюваність. Класи обчислювальних функцій.

ТЕМА 2. ПРЕДМЕТНІ ОБЛАСТІ ТА МЕТОДИ ЇХ ОПИСУ. МЕТОДИ РОЗРОБКИ ПРОГРАМ

Методи опису предметних областей. Методи специфікації вимог до програмних систем. Методології розробки програмних систем. Логічні формалізми специфікації програм. Класична та неklasична логіка для специфікацій програм.

ТЕМА 3. ПРИКЛАДИ СПЕЦИФІКАЦІЙ ПРОГРАМНИХ СИСТЕМ

Приклади мов специфікацій програм, їх порівняння. Програмні реалізації специфікацій програмного забезпечення, їх можливості для повноцінної розробки.

ЗМІСТОВНИЙ МОДУЛЬ 2. КОРЕКТНІСТЬ ПРОГРАМ ТА ФОРМАЛЬНА ВЕРИФІКАЦІЯ

ТЕМА 4. ТЕХНОЛОГІЇ ТА ІНСТРУМЕНТАЛЬНІ ЗАСОБИ СПЕЦИФІКАЦІЇ І РОЗРОБКИ ПРОГРАМ ЗА ДОПОМОГОЮ ЛОГІК ФЛОЙДА-ХОАРА

Можливості програмної реалізації специфікації та розробк програмних систем на базі логіки Флойда-Хоара. Огляд та порівняння програмного інструментарію специфікації програм на базі логіки Флойда-Хоара.

ТЕМА 5. ВЕРИФІКАЦІЯ СИСТЕМ В TLA, Z, B ТА RAISE

Верифікація програм з підтримкою паралельних обчислень. Проблеми специфікації та верифікації паралельних системи. Мова TLA, її можливості для специфікації та верифікації програм. Мова Z, її можливості, обмеження для специфікації та верифікації програмних систем. Програмна реалізація.

Мова B, її можливості, обмеження для специфікації та верифікації програмних систем. Інструментарій реалізації. Мова RAISE, її можливості, обмеження для специфікації та верифікації програмних систем. Інструментарій реалізації.

4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин											
	денна форма						Заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	сем	лаб	інд	с.р.		л	п	лаб	інд	с.р.
1	2	3	4	5	6	7	8	9	10	11	12	13
ЗМІСТОВНИЙ МОДУЛЬ 1. БАЗОВІ ПРИНЦИПИ ФОРМАЛЬНОГО ПІДХОДУ ДО РОЗРОБКИ ПРОГРАМ ТА СПЕЦИФІКАЦІЯ												
Тема 1. Мови специфікації. Формальні моделі обчислюваних функцій	15	1,5	3			10						
Тема 2. Предметні області та методи їх опису. Методи розробки програм.	15	2	3			10						
Тема 3. Приклади специфікацій програмних систем.	15	1,5	4			10						
Разом за змістовим модулем 1	45	5	10			30						
Тема 4. Технології та інструментальні засоби специфікації і розробки програм за допомогою логік Флойда-Хоара.	15	2	5			15						
Тема 5. Верифікація систем в TLA, Z, B та RAISE	15	3	5			5						
Разом за змістовим модулем 2	45	5	10			30						
Всього	90	10	20			60						

5. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Формальна специфікація конкретної програмної системи з використанням однієї з мов специфікацій (на вибір)	30
2	Верифікація специфікованої програмної системи з використанням відповідного методу верифікації та інструментарію	30
	Разом	60

6. Методи навчання.

- M1. Лекція (проблемна, інтерактивна)
- M3. Проблемне навчання
- M4. Проектне навчання (індивідуальне, малі групи, групове)
- M5. Онлайн навчання

7. Методи і форми контролю

Поточний контроль: усне опитування (індивідуальне, фронтальне, групове), комп'ютерне тестування, виконання розрахункових на комп'ютері згідно програми, захист робіт;

Підсумковий контроль: тестування (залік), опитування (співбесіда)

МК1. Тестування

МК2. Контрольне завдання

МК4. Методи усного контролю

МК6. Залік

МК7. Звіт.

9. Розподіл балів.

Оцінювання відбувається згідно положення «Про екзамени та заліки у НУБіП України»

Оцінка національна	Рейтинг, бали
Відмінно	90-100
Добре	74-89
Задовільно	60-73
Незадовільно	0-59

Для визначення рейтингу із засвоєння дисципліни $R_{\text{дис}}$ (до 100 балів) одержаний рейтинг з атестації (до 30 балів) додається до рейтингу з навчальної роботи $R_{\text{НР}}$ (до 70 балів): $R_{\text{дис}} = R_{\text{НР}} + R_{\text{АТ}}$

10. Рекомендована література

Основна

1. Нікітченко М.С. Теорія програмування: Частина 1.– Ніжин: Видавництво НДУ імені Миколи Гоголя, 2010.– 119 с.

2. Нікітченко М.С., Шкільняк С.С. Математична логіка та теорія алгоритмів. – К., 2008.

3. В. Boehm A spiral model of software development and enhancement. – Електронний ресурс. URL: <https://dl.acm.org/toc/sigsoft/1986/11/4>.

4. The RAISE specification language. Prentice Hall Int.– 1992.– 397 p.

5. Лавров С. 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.