

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Кафедра теорії та технології програмування



Заступник декана
з навчальної роботи

Олена КАШПУР

«12» лютого 2021 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
МЕТОДИ СПЕЦИФІКАЦІЇ ПРОГРАМ

для студентів

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

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

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

освітня програма **«Інформатика»**
(назва освітньої програми)

вид дисципліни **вибіркова**

вибірковий блок **«Теорія та технологія програмування»**

Форма навчання	денна
Навчальний рік	2022/2023
Семестр	8
Кількість кредитів ECTS	4
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	іспит

Викладач: к. ф.-м.н., 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 – 2021

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

ЗАТВЕРДЖЕНО

Зав. кафедри «Теорії та технології програмування»

(підпис)
ініціали)

Микола НІКІТЧЕНКО

(прізвище та

Протокол № 6 від «11» листопада 2021 р.

Схвалено гарантом освітньо-професійної програми «Інформатика»

Людмила ОМЕЛЬЧУК «11» листопада 2021 рік

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

Протокол від «11» листопада 2021 року № 7

Голова науково-методичної комісії

Людмила ОМЕЛЬЧУК

1. Мета дисципліни – знайомство з формальними методами специфікацій та засвоєння формальних методів специфікацій програм, їх практичне застосування на прикладі мов формальних специфікацій Z, Dafny, B та UML/OCL.

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

1. *Знати:* основні поняття теорії програмування, методи формалізації мов програмування, зокрема формалізації та аналізу семантики та синтаксису програм; теорію найменших нерухомих точок.

2. *Вміти:* формалізувати синтаксис мов програмування за допомогою БНФ та граматик, робити синтаксичний аналіз програм, будувати семантичний терм програми в алгебрі програм, доводити коректність програм. Формалізувати та досліджувати рекурсивні програми.

3. *Володіти елементарними навичками:* програмування сучасними мовами.

3. Анотація навчальної дисципліни:

Навчальна дисципліна “Методи специфікації програм” є складовою освітньо-професійної програми підготовки фахівців за першим (бакалаврським) рівнем вищої освіти галузі знань 12 „Інформаційні технології” зі спеціальності 122 „Комп’ютерні науки”, вибірковий блок „Теорія та технологія програмування” освітньо-професійної програми „Інформатика”.

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

Викладається у 8 семестрі 4 курсу в обсязі – 120 год.

(4 кредити ECTS) зокрема: лекції – 40 год, консультації – 2 год., самостійна робота – 78 год. У курсі передбачено 2 частини, 2 контрольні роботи та 1 проектна робота. Завершується дисципліна – іспитом в 8 семестрі.

В результаті вивчення навчальної дисципліни студент повинен:

знати архітектурні рішення і засоби проектування систем, вимоги до архітектури систем, стандарти моделювання, методи аналізу потреб, методи формальної специфікації програмних систем.

вміти розробляти формальні моделі ПС, розробляти шаблони і моделі, застосовувати моделі оцінки і параметри для аналізу вартості стосовно до фаз життєвого циклу програмного забезпечення, розробляти функціональні специфікації з урахуванням встановлених вимог.

Для допуску до дисципліни „Методи специфікації програм” спеціалізації «Теорія та технологія програмування» освітньо-професійної програми «Інформатика» студент повинен опанувати компетентності та результати навчання, які надає дисципліна „Теорія програмування” програми «Інформатика».

4. Завдання (навчальні цілі):

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

- здатність до математичного та логічного мислення, формулювання та досліджування математичних моделей, зокрема дискретних математичних моделей, обґрунтування

вибору методів і підходів для розв'язування теоретичних і прикладних задач у галузі комп'ютерних наук, аналізу та інтерпретування;

- здатність до алгоритмічного та логічного мислення;
- здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

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			
PH1.1	Знати архітектурні рішення і засоби проєктування систем та вимоги до архітектури систем, стандарти моделювання.	Лекція	Контрольна робота, 60% правильних відповідей, іспит	12%
PH1.2	Знати методи аналізу потреб, методи формальної специфікації програмних систем.	Лекція	Контрольна робота, 60% правильних відповідей, іспит	12%
PH2.1	Вміти розробляти формальні моделі ПС.	Лекція, самостійна робота	Контрольна робота, 60% правильних відповідей; проєктна робота; іспит	20%
PH2.2	Вміти розробляти шаблони і моделі, застосовувати моделі оцінки і параметри для аналізу вартості стосовно до фаз життєвого циклу програмного забезпечення	Лекція, самостійна робота	Контрольна робота, 60% правильних відповідей, іспит	10%
PH2.3	Вміти розробляти функціональні специфікації з урахуванням встановлених вимог	Лекція, самостійна робота	Контрольна робота, 60% правильних відповідей; проєктна робота; іспит	20%
PH3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проєктування та розробки архітектури програм, складати письмові звіти	Самостійна робота	Проєктна робота; поточне оцінювання	10%
PH4.1	Організувати свою самостійну роботу для досягнення результату	Самостійна робота	Проєктна робота; поточне оцінювання	8%
PH4.2	Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість	Самостійна робота	Проєктна робота; поточне оцінювання	8%

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 2.3	РН 3.1	РН 4.1	РН 4.2
Програмні результати навчання								
<i>(з опису освітньої програми)</i>								
ПРН17.2. Знати та вміти застосовувати підходи щодо оцінки та забезпечення якості програмного забезпечення.	+	+		+		+		+
ПРН19.2. Вибирати вихідні дані для проєктування, керуючись формальними методами опису вимог та моделювання.		+	+		+	+	+	

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

- семестрове оцінювання:

1. Контрольна робота 1: РН 1.1., РН 1.2 — 20 балів/12 балів.
2. Контрольна робота 2: РН1.2, РН 2.1, РН 2.2, РН2.3 - 20 балів/12 балів.
3. Проектна робота: РН2.1, РН 2.2, РН2.3, РН3.1, РН4.1, РН4.2 – 20 балів/12 балів.

- підсумкове оцінювання (у формі іспиту) вказується:

- максимальна кількість балів які можуть бути отримані студентом: 40 балів;
- результати навчання які будуть оцінюватись: РН1.1, РН1.2, РН2.1, РН2.2, РН2.3;
- форма проведення і види завдань: письмова.

Види завдань: 6 письмових завдань.

Критерії оцінювання на іспиті

Завдання	Тема завдання	Максимальний відсоток від 40 балів	Всього відсотків
Завдання 1,2	Письмові запитання з програмної інженерії та формальних методів	По 10%	20%
Завдання 3	Задання специфікації в мові Z (B)	25%	50%
Завдання 4	Задання специфікацій в мові Dafny	25%	
Завдання 5,6	Побудувати діаграму класів UML з обмеженнями OCL	По 15%	30%
			100%

Запитання для підготовки до іспиту

1. Формальна специфікація.
2. Метод формальної розробки ПС (формальний метод).
3. Визначення формальної нотації (формальної мови, мови специфікації).
4. Зв'язок та відмінності між мовами та методами формальної специфікації.
5. Підходи до методів формальної розробки.
6. Застосування специфікації програм на різних етапах життєвого циклу програми.
7. Класифікації специфікацій у відповідності до повноти опису.
8. Класифікації специфікацій у відповідності до об'єкту специфікації.
9. Класифікації методів специфікацій у відповідності до представлення моделі. Їх відмінності та спільні риси.
10. Класифікація мов специфікацій у відповідності до призначення та виду.

11. Класифікація мов специфікацій у відповідності до типу представлення специфікації.
12. Транзиційні специфікації (властивості, приклади).
13. Темпоральні (часові) специфікації (властивості, приклади).
14. Специфікації абстрактної моделі (властивості, приклади).
15. Алгебричні специфікації (властивості, приклади).
16. Аксиоматичні специфікації (властивості, приклади).
17. Рівні застосування формальних методів.
18. Основні поняття мови Z.
19. Множини в мові Z.
20. Способи задання множин в мові Z.
21. Кортежі в мові Z.
22. Схеми в мові Z. Об'єднання схем.
23. Видові схеми.
24. Приклади та типи параграфів в мові Z.
25. Вільні типи в мові Z.
26. Оцінювання Z специфікацій.
27. Декорування в мові Z та їх призначення.
28. Мова Dafny. Призначення та використання.
29. Основні конструкції специфікації у мові Dafny.
30. Перед- та постумови в мові Dafny.
31. Система типів мови Dafny.
32. Типи колекцій в мові Dafny.
33. Ідуктивні типи в мові Dafny.
34. Класи в мові Dafny.
35. Метод як основна одиниця мови Dafny.
36. Конструктивні рамки в мові Dafny.
37. Метрики завершення в мові Dafny.
38. Змінні, функції та предикати в мові Dafny.
39. Інваріант циклу в мові Dafny.
40. Доведення в мові Dafny. Логіка Хоара.
41. Мова UML/ OCL. Призначення та використання.
42. Діаграми мови UML. Діаграми класів.
43. Використання OCL для діаграм класів.
44. Основні типи мови об'єктних обмежень OCL.
45. Операції над типами мови об'єктних обмежень OCL.
46. Опис класу. Властивості класів в OCL
47. Вирази мови OCL.
48. Метод В. Основи. Призначення та застосування.
49. Концепції мови В. Абстрактна машина.
50. Абстрактна модель. Виконувана модель.
51. Композиційна семантика мови В.
52. Композиційна семантика узагальнених підстановок мови GSL та найслабших передумов.
53. Схема розробки програмної моделі В.

Студент не допускається до іспиту, якщо під час семестру набрав менше ніж 36 балів.

7.2 Організація оцінювання:

Терміни проведення форм оцінювання:

- 1. Контрольна робота 1: до 4 тижня семестру.*
- 2. Контрольна робота 2: до 15 тижня семестру.*
- 3. Проектна робота: до 15 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 разі неякісного виконання проектної роботи, викладач має право не зарахувати проектну роботу, або знизити за неї бали.

Студент має право здавати проектну роботу після закінчення визначеного для них терміну, але з втратою п'яти балів за кожен тиждень, який пройшов з моменту закінчення терміну її здачі.

7.3 Шкала відповідності оцінок

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

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

№ лекції	Назва лекції	Кількість годин		
		Лекції	Практичне заняття	Самостійна робота
Частина 1. Інженерія програмних систем. Огляд методів та мов формальних специфікацій програм				
1.	Тема 1. Інженерія програмних систем (ПС).	2		
2.	Тема 2. Основні етапи життєвого циклу (ЖЦ) ПС. Роль формальних специфікацій та методів аналізу в життєвому циклі розробки програмного забезпечення.	2		
3.	Тема 3. Призначення та застосування специфікацій програм. Поняття специфікації, мови специфікацій та методи формальної розробки.	2		
4.	Тема 4. Класифікація методів та мов формальних специфікацій програм: Специфікації абстрактної моделі. Алгебраїчні специфікації.	2		
5.	Тема 5. Класифікація методів та мов формальних специфікацій програм. Специфікації переходів станів. Аксиоматичні специфікації. Часові логічні специфікації.	2		
6.	Тема 6. Приклади специфікацій програмних систем із застосуванням різних методів специфікацій програмних систем. Рівні застосування формальних методів.	2		
	Самостійна робота. Стандартизація та сертифікація ПС.			12
	Контрольна робота 1	2		
	Всього по частині 1	14		12
Частина 2. Основні методи специфікації програм				
7.	Тема 7. Мова Z. Основні поняття мови Z. Синтаксис мови Z.	1		
8.	Тема 8. Мова Z. Типи мови Z. Множини та множина типів. Кортезжі та декартові добутки. Зв'язування та схема типів. Змінні та область їх дії в мові Z.	2		
9.	Тема 9. Мова Z. Властивості та схеми. Об'єднання властивостей схем. Об'єднання схем.	2		
10.	Тема 10. Мова Z. Відношення та функції в мові Z. Математичні засоби мови Z. Способи задання множин, відношень, функцій, натуральних чисел та послідовностей.	2		
11.	Тема 11. Мова Z. Оцінювання Z специфікації.	1		
	Самостійна робота. Опис програмної системи засобами мови Z.			4
12.	Тема 12. Мова Dafny. Призначення та використання Dafny. Основні конструкції специфікації у мові Dafny: перед- та післяумови; конструктивні рамки; метрики завершення; інваріанти.	2		

13.	Тема 13. Мова Dafny. Система типів мови Dafny. Скалярні типи; типи колекцій, кортежі, індуктивні типи; класи; масиви та типажі.	2		
14.	Тема 14. Мова Dafny. Метод як основна одиниця програми Dafny. Ghost-змінні, функції.	2		
15.	Тема 15. Мова Dafny. Доведення в мові Dafny. Логіка Хоара.	2		
	Самостійна робота. Опис програмної системи засобами Dafny.			4
16.	Тема 16. Мова UML\OCL. Призначення та використання мови моделювання UML та мови об'єктних обмежень OCL. Види діаграм UML. Діаграма класів,	2		
17.	Тема 18. Мова UML\OCL. Основні типи мови об'єктних обмежень OCL і операції над ними.	2		
18.	Тема 19. Мова UML\OCL. Властивості класів.	2		
19.	Тема 20. Мова UML\OCL. Вирази мови OCL.	2		
	Самостійна робота. Опис програмної системи засобами UML\OCL.			4
	Самостійна робота. Метод В для специфікації програм.			4
	Контрольна робота 2	2		
Всього по частині 2		26		16
Самостійна робота. Проектна робота «Розробка формальної моделі програмної системи з використанням мови Z/Dafny/B/UML(OCL)»				50
Консультація			2	
Іспит				
ВСЬОГО		40	2	78

Загальний обсяг 120 год., в тому числі:

Лекцій –40 год.

Консультації – 2 год.

Самостійна робота - 78 год.

9. Рекомендовані джерела:

Основні

1. SMT-LIB Theory of Nominative Data / Omelchuk L. and Shyshatska O. // Springer Nature Switzerland AG, Communications in Computer and Information Science, 1175 CCIS, pp. 89-110, 2020.
2. I. Rodhe, M. Karresand. Overview of formal methods in software engineering. – FOI Swedish Defence Research Agency, 2015. – 50 p.
3. S. Kumar, R.S. Suryavanshi, G. Chandra. Formal Methods: Techniques and Languages For Software Development // International Journal of Engineering Science and Advanced Research 1 (1), 2015. – P. 35-42.
4. Омельчук Л.Л. Формальні методи специфікації програм // К.: УкрІНТЕІ, 2009. – 78 с.

5. М.С. Нікітченко, Теорія програмування: Частина 1.– Ніжин: Видавництво НДУ імені Миколи Гоголя, 2010.– 119 с.
6. Лаврищева Е.М. Современные методы программирования: возможности и инструменты // Проблемы програмування. – 2006.– № 2-3. – С.60-74.
7. Буй Д.Б., Нікітченко М.С., Омельчук Л.Л., Редько В.Н., Шишацька О.В. Класифікація мов та методів формальних специфікацій програмних систем // Theoretical and applied aspects of program systems development (ТААПСД'2008). – Abstracts. – Berdyansk, Ukraine, 2007. P. 27-33.
8. Зубенко В.В. Омельчук Л.Л. Програмування // К.: ВПЦ "Київський університет", 2011.– 624 с.
9. Буч Г., Якобсон А., Рамбо Дж. UML. Классика CS. 2-е изд. / Пер. с англ.; Под общей редакцией проф. С. Орлова — СПб.: Питер, 2006. — 736 с. ISBN 5-469-00599-2
10. Крэг Ларман. Применение UML 2.0 и шаблонов проектирования = Applying UML and Patterns : An Introduction to Object-Oriented Analysis and Design and Iterative Development. — 3-е изд. — М.: Вильямс, 2006. — 736 с. — ISBN 0-13-148906-2
11. J. C. P. Woodcock and J. Davies. Using Z: Specification, Proof and Refinement. Prentice Hall International Series in Computer Science, 1996.

Додаткові:

1. Kumar, Subodh; Chandra, Girish; Yadav, Divakar : “Formal Verification of Security Protocol with B method”. Proceedings of the International Conference on Computer and Communication Technology (ICCCT), IEEE; (161-167), 2014
2. Leuschel, Michael and Butler, Michael. ProB: An Automated Analysis Toolset for the B Method // International Journal on Software Tools for Technology Transfer. 2008 – N.10, (2). – P.185-203.
3. Xiaoping Jia. ZTC: A Type Checker for Z Notation, User’s Guide, Version 2.03, June 1995. Available via anonymous ftp at ise.cs.depaul.edu.
4. Xiaoping Jia. An approach to animating Z specifications. In Proc. 19th Annual Int’l Computer Software and Applications Conf., Dallas, Texas, USA, August `1995.
5. http://fedoraproject.org/wiki/Formal_methods_tool_suite
6. http://en.wikipedia.org/wiki/Category:Formal_methods_tools
7. М.С. Нікітченко, С.С. Шкільняк. Математична логіка та теорія алгоритмів. – К., 2008.
8. J.M. Spivey. The Z Notation: A Reference Manual.–Oriel College, Oxford, OX1 4EW, England.– 1998.– 158 p.
9. Woodcock J.C.P., Davies J. Using Z: Specification, Refinement and Proof. – Prentice Hall, 1996. – 523 p.
10. Gary T. Leavens, Erik Poll, Curtis Clifton, Yoonsik Cheon, Clyde Ruby, David Cok, Peter Müller, Joseph Kiniry, Patrice Chalin, and Daniel M. Zimmerman. JML Reference Manual (DRAFT), September 2009.
11. <https://rise4fun.com/Dafny/tutorial/Guide>
12. http://www.doc.ic.ac.uk/~scd/Dafny_Material/Lectures.pdf
13. <http://citeserx.ist.psu.edu/viewdoc/download?doi=10.1.1.156.9592&rep=rep1&type=pdf>