

**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  
МЕТОДИ СПЕЦИФІКАЦІЇ ПРОГРАМ**

**для студентів**

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

Форма навчання

**денна**

Навчальний рік

**2018/2019**

Семестр

**8**

Кількість кредитів ECTS

**4**

Мова викладання, навчання

та оцінювання

**українська**

Форма заключного контролю

**екзамен**

Викладачі: **к.ф.-м.н., асистент Криволап А.В.**

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

на 20\_\_/20\_\_ н.р. \_\_\_\_\_ (\_\_\_\_\_) «\_\_» 20\_\_ р.  
(підпис, ПІБ, дата)

**КИЇВ – 2018**

Розробник: Криволап Андрій Володимирович, к.ф.-м.н., асистент кафедри «Теорії та технології програмування»

ЗАТВЕРДЖЕНО

В.о. Зав. кафедри «Теорії та технології програмування»

\_\_\_\_\_ (Панченко Т.В.)  
(підпис) (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 \_\_\_\_\_ (Хусаїнов Д.Я.)  
(підпис) (прізвище та ініціали)

«\_\_\_\_» \_\_\_\_\_ 20\_\_ року

**1. Мета дисципліни** – знайомство з формальними методами специфікацій та засвоєння формальних методів специфікацій програм, їх практичне застосування на прикладі мови формальних специфікацій Z та мови 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”, спеціалізації „Теорія та технологія програмування” освітньо-професійної програми „Інформатика”.

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

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

(4 кредити ECTS) зокрема: лекції – 44 год, консультації – 2 год., самостійна робота – 74 год. У курсі передбачено 3 змістових модулі та 3 модульні контрольні роботи. Завершується дисципліна – **екзаменом 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			
RH1.1	Знати архітектурні рішення і засоби проектування систем та вимоги до архітектури систем, стандарти моделювання.	Лекція	Контрольна робота, 60% правильних відповідей, екзамен	15%
RH1.2	Знати методи аналізу потреб, методи формальної специфікації програмних систем.	Лекція	Контрольна робота, 60% правильних відповідей, екзамен	15%
RH2.1	Вміти розробляти формальні моделі ПС.	Лекція, самостійна робота	Контрольна робота, 60% правильних відповідей, екзамен	30%
RH2.2	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, екзамен	10%
RH2.3	Вміти розробляти функціональні специфікації з урахуванням встановлених вимог	Лекція, самостійна робота	Контрольна робота, 60% правильних відповідей, екзамен	20%
RH3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування та розробки архітектури програм, скласти письмові звіти	Самостійна робота	Поточне оцінювання	5%
RH4.1	Організовувати свою самостійну роботу для досягнення результату	Самостійна робота	Поточне оцінювання	3%
RH4.2	Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість	Самостійна робота	Поточне оцінювання	2%

## 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>								
ВПРН2.1. Знати та вміти застосовувати підходи щодо оцінки та забезпечення якості програмного забезпечення.	+	+		+		+		+
ВПРН2.3. Вибирати вихідні дані для проектування, 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: РН2.1 - 20 балів/12 балів.
3. Контрольна робота 3: РН1.1, РН1.2, РН 2.2, РН2.3 – 20 балів/12 балів.

#### - підсумкове оцінювання (у формі екзамену) 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 екзамені

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 множини в мові Z	15%	40%
Завдання 4	Задання специфікацій в мові Z	25%	
Завдання 5,6	Побудувати діаграми UML	По 20%	40%
			<b>100%</b>

### Запитання для підготовки до екзамену

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 специфікацій (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. Множини в мові Z.
19. Способи задання множин в мові Z.
20. Кортежі в мові Z.
21. Схеми в мові Z.
22. Видові схеми.
23. Приклади та типи параграфів в мові Z.
24. Вільні типи в мові Z.
25. Оцінювання специфікацій.
26. Декорування в мові Z та їх призначення.
27. Мова UML.
28. Діаграми варіантів використання (діаграми прецедентів) та потоки подій.
29. Діаграми класів.
30. Використання OCL для діаграм класів.
31. Діаграми діяльності.
32. Діаграми взаємодії.
33. Діаграми станів.
34. Діаграми компонентів.
35. Діаграми розгортання.

*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 4 тижня семестру.*
- 2. Контрольна робота 2: до 9 тижня семестру.*
- 3. Контрольна робота 3: 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 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.

У разі неякісного виконання лабораторної роботи, викладач має право не зарахувати лабораторну роботу, або знизити за неї бали.

Студент має право здавати лабораторні роботи після закінчення визначеного для них терміну, але з втратою одного балу за кожен тиждень, який пройшов з моменту закінчення терміну її здачі.

### 7.3 Шкала відповідності оцінок

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

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

№ лекцій	Назва лекції	Кількість годин		
		Лекції	Практ. занять	Сам. р-та
<b>Частина 1. Інженерія програмних систем. Огляд методів та мов формальних специфікацій програм</b>				
1.	<b>Тема 1.</b> Інженерія програмних систем (ПС).	2		2
2.	<b>Тема 2.</b> Основні етапи життєвого циклу (ЖЦ) ПС. Роль формальних специфікацій та методів аналізу в життєвому циклі розробки програмного забезпечення.	2		4
3.	<b>Тема 3.</b> Призначення та застосування специфікацій програм. Поняття специфікації, мови специфікацій та методи формальної розробки.	2		4
4.	<b>Тема 4.</b> Класифікація методів та мов формальних специфікацій програм: Специфікації абстрактної моделі. Алгебраїчні специфікації.	2		4
5.	<b>Тема 5.</b> Класифікація методів та мов формальних специфікацій програм. Специфікації переходів станів. Аксиоматичні специфікації. Часові логічні специфікації.	2		4
6.	<b>Тема 6.</b> Приклади специфікацій програмних систем із застосуванням різних методів специфікацій програмних систем.	2		4
7.	<b>Тема 7.</b> Рівні застосування формальних методів.	2		2
	Контрольна робота 1			
Всього по частині 1		14		24
<b>Частина 2. Основні поняття мови Z</b>				
8.	<b>Тема 8.</b> Основні поняття мови Z.	1		4
9.	<b>Тема 9.</b> Синтаксис мови Z.	1		4
10.	<b>Тема 10.</b> Типи мови Z. Множини та множина типів. Кортежі та декартові добутки.	2		2
11.	<b>Тема 11.</b> Зв'язування та схема типів. Змінні та область їх дії в мові Z.	2		2
12.	<b>Тема 12.</b> Властивості та схеми. Об'єднання властивостей схем. Об'єднання схем.	2		2
13.	<b>Тема 13.</b> Відношення та функції в мові Z. Математичні засоби мови Z.	2		4
14.	<b>Тема 14.</b> Способи задання множин, відношень, функцій, натуральних чисел та послідовностей.	2		4
15.	<b>Тема 15.</b> Оцінювання Z специфікації.	2		4
	Контрольна робота 2			
Всього по частині 2		14		26
<b>Частина 3. Уніфікована мова моделювання UML/OCL</b>				
16.	<b>Тема 16.</b> Призначення та використання UML. Види	2		2



	<i>діаграм UML.</i>			
17.	<b>Тема 17.</b> Діаграми UML та їх використання: діаграма прецедентів, діаграма класів, VOPC-діаграма класів.	2		2
18.	<b>Тема 18.</b> Діаграми UML та їх використання: діаграми реалізації (діаграма компонентів та діаграма розгортання).	2		4
19.	<b>Тема 19.</b> Діаграми UML та їх використання: діаграми поведінки (діаграма станів, діаграма діяльності), діаграми взаємодії (діаграма послідовності та діаграма кооперації).	2		4
20.	<b>Тема 20.</b> Призначення мови об'єктних обмежень UML/OCL.	2		4
21.	<b>Тема 21.</b> Основні типи мови об'єктних обмежень OCL і операції над ними.	2		4
22.	<b>Тема 22.</b> Властивості класів.	2		2
23.	<b>Тема 23.</b> Вирази мови OCL.	2		2
	<i>Контрольна робота 3</i>			
	Всього по частині 3	16		24
	Консультація		2	
	Екзамен			
	<b>ВСЬОГО</b>	<b>44</b>	<b>2</b>	<b>74</b>

**Загальний обсяг 120 год.,** в тому числі:

Лекцій – **44 год.**

Консультації – **2 год.**

Самостійна робота - **74 год.**

**Теми, винесені на самостійне вивчення:**

Застосування шаблонів проектування та UML для об'єктно-орієнтованого моделювання ПС.

Функціональне програмування.

Логічне програмування. Методи і алгоритми обчислень. Структури даних і механізми управління.

**9. Рекомендовані джерела:**

**Основна**

1. Омельчук Л.Л. Формальні методи специфікації програм // К.: УкрІНТЕІ, 2009. – 78 с.
2. М.С. Нікітченко, Теорія програмування: Частина 1.– Ніжин: Видавництво НДУ імені Миколи Гоголя, 2010.– 119 с.
3. Лаврищева Е.М. Современные методы программирования: возможности и инструменты // Проблемы програмування. – 2006.– № 2-3. – С.60-74.
4. 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 (TAAPSD'2008). – Abstracts. – Berdyansk, Ukraine, 2007. P. 27-33.
5. Зубенко В.В. Омельчук Л.Л. Програмування // К.: ВПЦ "Київський університет", 2011.- 624 с.
6. 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

7. 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

***Додаткова:***

8. 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](http://ise.cs.depaul.edu).

9. Xiaoping Jia. An approach to animating Z specifications. In Proc. 19th Annual Int'l Computer Software and Applications Conf., Dallas, Texas, USA, August `1995.

10. [http://fedoraproject.org/wiki/Formal\\_methods\\_tool\\_suite](http://fedoraproject.org/wiki/Formal_methods_tool_suite)

11. [http://en.wikipedia.org/wiki/Category:Formal\\_methods\\_tools](http://en.wikipedia.org/wiki/Category:Formal_methods_tools)

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

13. J.M. Spivey. The Z Notation: A Reference Manual.—Oriel College, Oxford, OX1 4EW, England.— 1998.— 158 p.

14. Woodcock J.C.P., Davies J. Using Z: Specification, Refinement and Proof. – Prentice Hall, 1996. – 523 p.

15. 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.