

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ  
Кафедра теорії та технології програмування



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ  
ТЕОРІЯ І ТЕХНОЛОГІЯ РОЗРОБКИ ПРОГРАМНИХ  
СИСТЕМ /  
THEORY AND TECHNOLOGY OF SOFTWARE SYSTEMS  
DEVELOPMENT

для здобувачів освітньо-наукового рівня «доктор філософії»/  
for obtaining an educational-scientific degree: Doctor of Philosophy

галузь знань	12 «Інформаційні технології»/ “Information Technologies”
спеціальність	121 «Інженерія програмного забезпечення» / “Software Engineering”
освітній рівень	третій (освітньо-науковий)/ third (educational-scientific level)
освітньо-наукова програма	«Інженерія програмного забезпечення» / “Software Engineering”
вид дисципліни	обов'язкова / mandatory
Форма навчання	денна / заочна
Навчальний рік	20 <u>18</u> / 20 <u>19</u>
Рік навчання	1
Кількість кредитів ECTS	3
Мова викладання, навчання та оцінювання	англійська/English
Форма заключного контролю	екзамен/exam

Викладачі: професор Нікітченко Микола Степанович, д.ф.-м.н.

Пролонговано: на 20 19 / 20 20 н.р. ( ) «15» 04 2019 р.  
на 20 / 20 н.р. ( ) « » 20 р.

КИЇВ – 20 18

Розробник: **Нікітченко Микола Степанович**, д. ф.-м. н., проф., завідувач кафедри теорії та технології програмування

ЗАТВЕРДЖЕНО

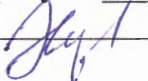
Завідувач кафедри теорії та технології програмування

  
\_\_\_\_\_  
(підпис) Нікітченко М.С..

Протокол №4 від «19» 12 2017р.

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

Протокол від «14» 02 2018 року № 6

Голова науково-методичної комісії   
\_\_\_\_\_  
(підпис) професор, д.ф.-м.н. Хусайнов Д.Я.

**1. Мета дисципліни** дати сучасні знання про теоретичні основи та технології побудови програмних систем, розвинути здатності формулювати наукову проблему та робочі гіпотези на основі глибокого осмислення наявних і створення нових цілісних знань, а також професійної практики, розвивати й реалізовувати нові конкурентоздатні ідеї в галузі інформаційних технологій.

/

**Discipline aim.** The aim of the discipline is to provide up-to-date knowledge of theoretical foundations and technologies of software systems construction, to develop ability to formulate scientific problem and working hypotheses on the basis of deep understanding of existing and creation of new holistic knowledge, as well as professional practice, to develop and implement new competitive ideas in the field of information technology.

## **2. Попередні вимоги до опанування або вибору навчальної дисципліни/:**

### **Preliminary demands to master or choice of the course discipline:**

1. *Знати:* основні методи формалізації предметних областей та програм, сучасні засоби розробки та верифікації програмних систем.
2. *Вміти:* розробляти, аналізувати та застосовувати програмні системи для розв'язання завдань та прикладних задач, використовуючи сучасні методи розробки програм.

/

1. To know: basic methods of subject domain formalisation, modern methods of development and verification of software systems.

2. To be able to: develop, analyse and apply software systems to solve problems and applied tasks using modern software development methods.

## **3. Анотація навчальної дисципліни / Synopsis of the course:**

Дисципліна «Теорія і технологія розробки програмних систем» належить до переліку обов'язкових навчальних дисциплін. Вона забезпечує професійний розвиток аспіранта, спрямована на формування теоретичних основ та новітніх технологій розробки програмних систем, зокрема, розуміння методів їх формалізації, специфікації та верифікації на основі логіко-алгебраїчного підходу, а також вміння використовувати сучасні засоби підтримки розробки інформаційних систем, засновані на програмних логіках.

/

The discipline "Theory and Technology of Software Systems Development" belongs to the list of mandatory discipline. It provides the professional development of PhD students, aimed at the formation of theoretical foundations and the latest technologies of software systems development, in particular, understanding of methods of their formalisation, specification and verification on the basis of logic-algebraic approach, as well as ability to use modern tools of development of information systems based on software logic.

**4. Завдання (навчальні цілі)/ Objectives of study:** набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень у програмуванні, відповідно науково-освітньої кваліфікації «Доктор філософії». Зокрема, розвивати:

- здатність генерувати нові ідеї (креативність);
- вміння виявляти, ставити і вирішувати проблеми в галузі інформаційних технологій;
- здатність накопичувати, обробляти та систематизувати професійні знання в інформаційних технологіях і визнавати важливість навчання протягом всього життя;
- здатність розвивати й реалізовувати нові конкурентоздатні ідеї в галузі інформаційних технологій;

- здатність застосовувати і розвивати фундаментальні і міждисциплінарні знання, включаючи математичні і наукові принципи, методи, засоби та нотації для успішного розв’язання проблем;
- розуміння теоретичних засад, що лежать в основі методів досліджень інформаційних систем та програмного забезпечення, методології проведення досліджень та обчислювальних експериментів.

/

Objectives (learning objectives): acquiring knowledge, skills and competences at the level of the latest achievements in programming, according to the scientific and educational qualification of “Doctor of Philosophy”. In particular, to develop:

- the ability to conduct research at an appropriate level;
- ability to search, process and analyse information from various sources;
- the ability of accumulating, understanding and systematizing the professional knowledge of information technologies and awareness of the importance of lifelong learning;
- the ability to develop and implement new competitive ideas in the field of information technologies;
- the ability of implementation and development of fundamental and interdisciplinary knowledge, including mathematical and scientific principles, methods, tools, and notations for successful solving of problems;
- understanding of theoretical foundations that are in the basis of investigation of information technologies and software systems, and computational experiments, methodology of the implementation of the environmental and enumeration experiments.

## 5. Результати навчання за дисципліною/ Results of learning:

Результат навчання (РН) (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			
РН 1.1	Знати основні методи формалізації програмних систем (програмні алгебри, мови специфікацій, програмні логіки)/ To know main methods of program system formalization (program algebras, specification languages, program logics)	<i>Лекція/ Lecture</i>	<i>Контрольна робота 1 (60% правильних відповідей), екзамен, активна робота на лекції, усні відповіді/ Test 1 (60% correct answers), exam, activity during lectures, oral answers</i>	20%
РН 1.2	Знати основні структури даних, функції та композиції, які використовуються в мовах розробки програмних систем / To know main structures of data, functions, and compositions, used in software development languages			
РН 1.3	Знати методи послідовного уточнення програм / To know methods of successive program refinements			20%
РН 1.4	Знати методи доведення правильності побудови програм / To know methods of program development verification			
РН 2.1	Вміти застосовувати методи формалізації предметної області / Be able to apply subject domain formalization methods	<i>Лекція, самостійна робота/ Lecture, Individual work</i>	<i>Контрольна робота 2 (60% правильних відповідей), екзамен, захист проекту, виконання завдань, винесених на самостійну</i>	20%
РН 2.2	Вміти аналізувати функціональні вимоги до програмних систем / Be able to analyze functional requirements for software systems			20%

			<i>роботу/</i> Test 2 (60% correct answers), exam, project defense, accomplishment of tasks assigned to <i>Individual work</i>	
PH 2.3	Вміти застосовувати програмні засоби розробки систем / Be able to use systems development tools	<i>Самостійна робота/ Individual work</i>	<i>Захист проекту/ project defense</i>	5%
PH3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування та розробки програм, складати письмові звіти / Be able to justify own view of the problem, communicate with colleagues in the design and development of programs, prepare written reports			5%
PH4.1	Демонстрація авторитетності, інноваційності, високого ступеня самостійності, академічної та професійної доброчесності, послідовної відданості розвитку нових ідей або процесів у передових контекстах професійної та наукової діяльності, що стосується теорії та технології програмування. / Demonstration of authority, innovativeness, high degree of independence, academic and professional integrity, consistent dedication to the development of new ideas or processes in advanced contexts of professional and scientific activity related to the theory and technology of programming.			5%
PH4.2	Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість / Responsibly treat the works performed, be responsible for their quality			5%

## 6. Співвідношення результатів навчання дисципліни із програмними результатами навчання / Correspondence between learning results and program study results

Програмні результати навчання	Результати навчання дисципліни									
	PH 1.1	PH 1.2	PH 1.3	PH 1.4	PH 2.1	PH 2.2	PH 2.3	PH 3.1	PH 4.1	PH 4.2
<i>(з опису освітньої програми)</i>										
ПРН-8. Оцінювати, класифікувати і обґрунтовувати вибір методів формування вимог до інформаційної системи, формулювати вимоги / To evaluate, classify and justify the choice of methods for forming requirements for information systems, formulate requirements	+							+	+	+
ПРН-11. Розробляти засоби реалізації інформаційних технологій (методичні, інформаційні, математичні, алгоритмічні,		+	+	+			+			+

технічні і програмні) / To develop means of realizing of information technologies (methodical, information, mathematical, algorithmic, technical, and software)										
ПРН-12. Здійснювати аналітичне дослідження робочих параметрів інформаційних технологій, а також здійснювати аналіз вибраних методів, засобів реалізації проектування і давати їм критичну оцінку / To make analytical study of the operating parameters of information technologies, as well as analysis of the selected methods, implementation tools and give them a critical assessment						+	+	+		+
ПРН-13. Оцінювати і вибирати методи і моделі створення, впровадження, експлуатації інформаційних систем і керування ними на всіх етапах життєвого циклу / To evaluate and select methods and models for creating, implementing, operating and managing information systems at all stages of the life cycle	+									+

## 7. Схема формування оцінки.

### 7.1. Форми оцінювання здобувачів освітньо-наукового ступеня:

#### - оцінювання впродовж навчального періоду:

1. Активна робота на лекції, усні відповіді: РН1.1, РН1.2, РН1.3, РН1.4 – 5 балів/3 бали;
2. Виконання завдань, винесених на самостійну роботу: РН2.1, РН2.2 – 5 балів/3 бали;
3. Контрольна робота 1: РН1.1, РН1.2, РН2.1, РН2.2 – 15 балів/9 балів;
4. Контрольна робота 2: РН1.3, РН1.4, РН2.1, РН2.2 – 15 балів/9 балів;
6. Захист проекту: РН2.1, РН2.2, РН2.3, РН3.1, РН4.1, РН4.2, – 20 балів/12 балів;

#### - підсумкове оцінювання: екзамен.

- максимальна кількість балів які можуть бути отримані: 40 балів;
- результати навчання які будуть оцінюватись: РН1.1, РН1.2, РН1.3, РН1.4;
- форма проведення і види завдань: письмова робота.

Для здобувачів освітньо-наукового ступеня, які набрали сумарно меншу кількість балів ніж критично-розрахунковий мінімум – 20 балів для одержання іспиту за рішенням кафедри не допустити до складання іспиту із рекомендацією здати контрольні роботи та захистити проект до повторного складання іспиту.

Рекомендований мінімум – 36 балів.

### 7.2. Організація оцінювання:

Обов'язковим є виконання завдань, винесених на самостійну роботу, та модульних контрольних робіт за графіком робочої програми.

У частину 1 входять теми 1 - 3, у частину 2 – теми 4 – 6 у частину 3 – теми 7 – 9. Обов'язковим для екзамену є виконання усіх контрольних робіт та захист проекту до вказаної викладачем дати, перед початком екзаменаційної сесії, згідно навчального плану. Переписування чи перескладання

тем не практикується. Дозволяється здача окремих завдань модульних тем у проміжках між написанням модульних контрольних робіт (наприклад, перша тема здається до здачі наступної модульної контрольної роботи у будь-який зручний для викладача та студента час).

**Терміни проведення форм оцінювання:**

1. *Контрольна робота: до 5 тижня навчального періоду.*
2. *Контрольна робота: до 13 тижня навчального періоду.*
3. *Захист проекту: до 10 тижня навчального періоду.*

У випадку відсутності з поважних причин відпрацювання та перездачі контрольні роботи здійснюються у відповідності до „Положення про організацію освітнього процесу”.

### **7.3. Шкала відповідності оцінок**

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

**При визначені оцінки визначальною є робота в семестрі.** Після завершення розгляду тем проводяться письмові контрольні роботи та теоретичне опитування.

## 8. СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ. ТЕМАТИЧНИЙ ПЛАН ЛЕКЦІЙ

№	Назва лекції	Кількість годин		
		Лекції	Практичні	Самостійна робота
<b>Частина 1. «Основні методи формалізації програмних систем»</b> <b>Part 1. “Main methods of formalization of software systems”</b>				
1	<p><b>Тема 1.</b> Вступ, основні означення та поняття. <i>Самостійна робота:</i> На прикладі задач по темі дисертаційного дослідження провести аналіз використовуваних алгоритмів. /</p> <p><b>Theme 1.</b> Introduction, basic definitions and notions. <i>Individual work:</i> Analyze the algorithms used in a dissertation research.</p>	1		6
2	<p><b>Тема 2.</b> Багатоосновні алгебри. Моделювання предметної області за допомогою формалізму багатоосновних алгебр. <i>Самостійна робота:</i> На прикладі задач по темі дисертаційного дослідження провести побудову моделей предметної області. /</p> <p><b>Theme 2.</b> Multisorted algebras. Modeling a subject domain using the formalism of multisorted algebras. <i>Individual work:</i> Construct models of subject domains used in a dissertation research.</p>	1	2	12
3	<p><b>Тема 3.</b> Програмні алгебри та програмні логіки. <i>Самостійна робота:</i> Сформулювати по темі дисертаційного дослідження опис систем в програмній алгебрі. /</p> <p><b>Theme 3.</b> Program algebras and program logics. <i>Individual work:</i> Formulate in a program algebra the description of systems used in a dissertation research</p>	1		6
4	<p><b>Тема 4.</b> Методи аналізу програмних систем. /</p> <p><b>Theme 4.</b> Methods of analysis of program systems.</p>	1		12
<i>Контрольна робота 1 /</i> <i>Task 1.</i>		1		
<b>Частина 2. «Основні методи та технології розробки програмних систем»/</b> <b>Part 2. “Main methods and technologies for program systems development”</b>				
5	<p><b>Тема 5.</b> Методи специфікації програмних систем (B, TLA, Dafny). <i>Самостійна робота:</i> Написати специфікацію фрагмента програмної системи, пов'язаної із темою дисертаційного дослідження, за допомогою однієї з мов специфікацій.</p>	1	2	12



	<p>/</p> <p><b>Theme 5.</b> Methods of program systems specification (B, TLA, Dafny).</p> <p><i>Individual work:</i></p> <p>Using a specification language write a specification of a fragment of program system related to the topic of a dissertation research.</p>			
6	<p><b>Тема 6.</b> Формалізація вимог до програмних систем за допомогою програмних логік.</p> <p><i>Самостійна робота:</i></p> <p>Сформулювати по темі дисертаційного дослідження вимоги до фрагмента системи в програмній алгебрі.</p> <p>/</p> <p><b>Theme 6.</b> Formalization of requirements to program systems using program logic.</p> <p><i>Individual work:</i></p> <p>Formulate requirements in program logic to a fragment of software system used in a dissertation research.</p>	1		6
7	<p><b>Тема 7.</b> Методи верифікації програмних систем з використанням програмних логік.</p> <p><i>Самостійна робота:</i></p> <p>Верифікувати фрагмент програмної системи, пов'язаної із темою дисертаційного дослідження, з використанням програмної логіки.</p> <p>/</p> <p><b>Theme 7.</b> Methods of verification of program systems using program logic.</p> <p><i>Individual work:</i></p> <p>Verify with program logic a fragment of software system used in a dissertation research.</p>	1	2	12
8	<p><b>Тема 8.</b> Перспективні напрями створення методів та технологій розробки програмних систем.</p> <p>/</p> <p><b>Theme 8.</b> Prospective directions of creation of methods and technologies of software systems development.</p>	1		6
<i>Контрольна робота 2</i> <i>Task 2.</i>		1		
<b>ВСЬОГО/ TOTAL</b>		10	6	72

**Загальний обсяг 90 годин**, в тому числі / **Total duration 90 hours**, namely:

Лекцій / Lectures – **10 годин / hours**,

Практичні /Tutorials – **6 годин / hours**,

Консультації / Consultations - **2 години / hours**.

Самостійна робота / Individual work – **72 години / hours**.

## 9. Рекомендовані джерела

### *Основні / Main:*

1. С.Л. Кривий. Вступ до методів створення програмних систем. Київ, НаУКМА, 2018.–449.

2. Дорошенко А.Ю., Жереб К.А., Иванов Е.В., Никитченко Н.С., Яценко Е.А. Формальные методы построения параллельных программ, Кропивницкий, 2016.– 440 с.

*Додаткові / Additional:*

3. Bjorner D. Software Engineering 1: Abstraction and Modelling. Berlin: Springer, 2006.– 714 p.
4. Bjorner D. Software Engineering 2: Specification of Systems and Languages. Berlin: Springer, 2006.– 780 p.
5. Bjorner D. Software Engineering 3: Domains, Requirements, and Software Design. Berlin: Springer, 2010.– 768 p.
6. Abrial J.A. Assigning Programs to Meanings. Cambridge: Cambridge University Press, 2005.– 816 p.
7. Lamport L. Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers. Addison-Wesley Professional, 2002.– 384 p.
8. Leino, R. Dafny: An Automatic Program Verifier for Functional Correctness. Proceedings of the Conference on Logic for Programming, Artificial Intelligence, and Reasoning. 2010, pp. 348–370.