

# Практичний семінар з математичних методів в інженерії програмного забезпечення

## СИЛАБУС

Шифр і назва спеціальності	121 “Інженерія програмного забезпечення”	Інститут / факультет	КН
Назва програми	«Інженерія програмного забезпечення»	Кафедра	ПІІТУ
Тип програми	Освітньо-професійна	Мова навчання	українська, англійська

## Викладач

Хацько Наталія Євгенівна, [nataliia.khatsko@khp.edu.ua](mailto:nataliia.khatsko@khp.edu.ua)



К.т.н., доцент кафедри програмної інженерії та інформаційних технологій управління, доцент.

Підготувала і опублікувала понад 45 наукових та навчально-методичних публікацій.

SCOPUS Author ID <https://www.scopus.com/authid/detail.uri?authorId=57200820629>; Researcher ID

<https://app.webofknowledge.com/author/#/record/17252627>; Google Scholar

<https://scholar.google.com.ua/citations?user=US7Ovx4AAAAJ&hl=uk>; ORCID [orcid.org/0000-0002-2543-0280](https://orcid.org/0000-0002-2543-0280)

Основні курси: «Комп'ютерна математика (частини 1, 2, 3)», «Практичний семінар з математичних методів в інженерії програмного забезпечення», «Формальні методи верифікації програмних систем», «Формальні методи дослідження програмних систем».

## Загальна інформація про курс

Анотація	Дисципліна «Практичний семінар з математичних методів в інженерії програмного забезпечення» є навчальною дисципліною з циклу професійної обов'язкової підготовки за спеціальністю 121 “Інженерія програмного забезпечення”. Вона викладається у шостому семестрі в обсязі 150 год (5 кредитів ECTS), зокрема: практичні заняття – 32 год., самостійна робота – 118 год. У курсі передбачено виконання розрахункової роботи. Завершується дисципліна доповіддю щодо результатів виконання розрахункової роботи та заліком.
Цілі курсу	Формування у студентів практичних знань та умінь, які необхідні для застосування математичних методів в проектуванні програмних систем. Набуття практичних навичок щодо використання формальних методів і моделей дискретної математики при обробці дискретної інформації та опису дискретних процесів, пов'язаних з розробкою програмного забезпечення.
Формат	Міні-лекції, практичні роботи, самостійна робота, виконання розрахункової роботи. Підсумковий контроль — залік.
Семестр	6

Обсяг (кредити) / Тип курсу  
(обов'язковий / вибірковий)

5 / обов'язковий

Лекції (години)

-

Практичні заняття (години)

32

Самостійна робота (години)

118

Програмні компетентності

K01. Здатність до абстрактного мислення, аналізу та синтезу.

K02. Здатність застосовувати знання у практичних ситуаціях.

K03. Здатність спілкуватися державною мовою як усно, так і письмово.

K05. Здатність вчитися і оволодівати сучасними знаннями.

K06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

K22. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.

K26. Здатність до алгоритмічного та логічного мислення.

Результати навчання	Методи викладання та навчання	Форми оцінювання (поточне оцінювання CAS, підсумкове оцінювання FAS)
<p>PRO1. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.</p> <p>PR11 - Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.</p>	<p>Міні-лекції, практичні заняття у вигляді семінарів за відповідними темами, виконання індивідуального завдання у вигляді курсової роботи, самостійна робота з літературними джерелами.</p>	<p><b>Поточне оцінювання CAS:</b> Оцінювання роботи студентів у процесі практичних занять Проміжне оцінювання виконання складових частин розрахункової роботи.</p> <p><b>Підсумкове оцінювання FAS:</b> залік</p>

### СИСТЕМА ОЦІНЮВАННЯ

Розподіл балів для оцінювання успішності аспіранта	Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	Нарахування балів
	90-100	A	відмінно	
	82-89	B	добре	
	74-81	C		
	64-73	D	задовільно	
	60-63	E		
	35-59	FX		
	0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	

**100% підсумкове оцінювання у вигляді заліку (30%) та поточного оцінювання (70%).**  
**30% залік: ...**  
**70% поточне оцінювання: ...**  
 оцінювання завдань на практичних роботах;  
 Практичне заняття 1-2 10%  
 Практичне заняття 3-5 10%  
 Практичне заняття 6-7 10%  
 Практичне заняття 8-9 10%  
 Практичне заняття 10-11 10%  
 Практичне заняття 12-13 10%  
 Практичне заняття 14-15 10%

**Політика курсу** Студенти повинні відвідувати всі заняття відповідно до розкладу навчання та дотримуватись норм академічної етики. Студенти повинні працювати з обов'язковим та рекомендованим читанням, включаючи ресурси Інтернету. Студенти повинні виконати та подати всі домашні роботи протягом семестру, в якому викладається курс, до екзаменаційної сесії. Підсумкове оцінювання не проводиться без особистої присутності студентів.

### Структура та зміст курсу

Лекція 1	Практичне заняття 1-2	Інша
	<p>Аналіз первинного опису програмної системи, що планується до розробки. Приклад використання апарату скінченний автоматів при підготовці до</p>	<p>Обрати тему індивідуального індивідуальної розрахункової роботи, в рамках якої будуть застосовані математичні методи дослідження. Дослідження публікацій в наукових журналах, матеріалах технічних конференцій; в спеціалізованих ІТ-пабліках</p>

			розробки програмної системи.		
<b>Лекція 2</b>		<b>Практичне заняття 3-5</b>	Складання протоколу роботи програмної системи. Математичний опис скінченого автомата.		Проаналізувати чи вирішувалась раніше обрана тема за допомогою програмної реалізації. Розробка протоколів реакції сутностей на події системи. Виділення структурних дієвих сутностей, визначення системних подій та їх ініціаторів. Створення математичного опису скінченого автомата для кожної сутності (декомпозиція системи)
<b>Лекція 3</b>		<b>Практичне заняття 6-7</b>	Аналіз роботи скінченого автомату кожної сутності системи.		Аналіз предметної області роботи щодо виділених сутностей. Аналіз роботи скінченого автомату кожної сутності системи. Внесення змін до протоколів роботи за необхідністю.
<b>Лекція 4</b>		<b>Практичне заняття 8-9</b>	Автомат-добуток як опис системи в цілому.		Аналіз предметної області роботи щодо можливих реакцій на події в системі. Виконання з'єднання скінчених автоматів окремих сутностей. Аналіз можливих спільних реакцій при розгляді автомату-добутку. Внесення змін до протоколів за необхідністю.
<b>Лекція 5</b>		<b>Практичне заняття 10-11</b>	Опис мови, що допускається кожним з отриманих автоматів.		Виявлення можливих ланцюжків в роботі скінчених автоматів системи. Аналіз відповідності отриманих ланцюжків з протоколами системи.
<b>Лекція 6</b>		<b>Практичне заняття 12-13</b>	Регулярні вирази. Таблиці побудови регулярних виразів.		Отримання регулярних виразів щодо мови скінчених автоматів за допомогою таблиць для побудови регулярних виразів.
<b>Лекція 7</b>		<b>Практичне заняття 14-15</b>	Конструювання граматик.		Конструювання граматик.
<b>Лекція 8</b>		<b>Практичне заняття 16</b>	Доповідь щодо результатів виконання індивідуальної розрахункової роботи		Виконання розрахункової роботи.

### Література

1. Ajit Singh Formal Language And Automata Theory. (2019) LAMBERT Academic Publishing,
2. Abejide Ade-Ibijola New Finite Automata Applications in Novice Program Comprehension. (2017) LAP LAMBERT Academic Publishing,
3. Neeru Gupta (2020) Beginner's Guide - Automata Theory.
4. Ezhilarasu Umadevi Palani (2019) Finite Automata Problems & Solutions. - LAP Lambert Academic Publishing,
5. Stoyan Mihov, Klaus U. Schulz (2019) Finite-State Techniques: Automata, Transducers and Bimachines. - Cambridge University Press,...
6. Hopcroft, John E., Motwani, Rajeev (2014) Ullman, Jeffrey D. Introduction to automata theory, languages, and computation Boston : Pearson Education, Inc.
7. Хаггарти Р. (2017) Дискретная математика для программистов: учебное пособие. Москва: Техносфера,

1. Michael Sipser (2006) Introduction to the Theory of Computation - Thomson,
2. Daniel I. A. John C Martin, (1996) Cohen Introduction to Computer Theory, /
3. Introduction to languages and the Theory of Computation ,
4. Lewis H.P. & Papadimitiou C.H. Pearson Elements of Theory of Computation",
5. Theory of Computer Science and Automata languages and computation -Mishra and Chandrashekar, 2nd edition,
6. Foster, E. C. (1989) Software Engineering : A Methodical Approach. New York: Apress.
7. J. Richard Büchi Finite Automata, Their Algebras and Grammars. Towards a Theory of Formal Expressions.
8. Goswami D., Krishna K. V. (2010) Formal Languages and Automata Theory.
9. Завалишин Е.П. (2007) Логика. Учебное пособие для вузов. Тула : Изд-во ТулГУ
10. Gerda Ivanickienė. The theoretical material and exercises

#### ІНФОРМАЦІЙНІ РЕСУРСИ В ІНТЕРНЕТІ

1. D. Goswami and K. V. Krishna. Formal Languages and Automata Theory: <http://www.iitg.ernet.in/dgoswami/Flat-Notes.pdf>
2. Introduction to Automata and Complexity Theory: <http://infolab.stanford.edu/~ullman/ialc/spr10/spr10.html>
3. Formal Languages and Automata Theory: <http://cs.fit.edu/~dmitra/FormaLang>.

#### Норми академічної етики

Студент повинен дотримуватися «Кодексу етики академічних взаємовідносин та доброчесності НТУ «ХПІ»: виявляти дисциплінованість, вихованість, доброзичливість, чесність, відповідальність

Силабус за змістом повністю відповідає робочій програмі курсу.