

**НАЦІОНАЛЬНА АКАДЕМІЯ НАУК УКРАЇНИ
ІНСТИТУТ ПРИКЛАДНОЇ МАТЕМАТИКИ І МЕХАНІКИ**

«ЗАТВЕРДЖУЮ»

Директор ІПММ НАН України
академік НАН України



_____ Ігор СКРИПНІК
« 25 » вересня 2025 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

**Технології створення програмних продуктів.
Компонентне програмування**

**підготовки здобувачів ступеня вищої освіти
треть (освітньо-науковий)
доктора філософії**

галузь знань 11 Математика та статистика

спеціальність 113 Прикладна математика

Робоча навчальна програма навчальної дисципліни «Технології створення програмних продуктів. Компонентне програмування» для здобувачів наукового ступеня доктора філософії за спеціальністю 113 Прикладна математика.

Розробник:

Сенченко Олексій Сергійович, старший науковий співробітник відділу теорії керуючих систем, кандидат фізико-математичних наук, доцент.

Робочу програму схвалено на засіданні відділу теорії керуючих систем
Протокол № 4 від 18 вересня 2025 року.

Завідувач відділу



Кононов Ю.М.

Схвалено Науково-методичною радою Інституту прикладної математики і механіки НАН України.

Протокол № 4 від 22 вересня 2025 року.

Голова науково-методичної ради



Несмелова О.В.

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни
Кількість кредитів - 4	Галузь знань: 11 Математика та статистика	дисципліна вільного вибору аспіранта
Загальна кількість годин – 120	Спеціальність: 113 Прикладна математика	Рік підготовки:
		1-2
		Семестр
		2, 3
		Лекції
Тижневих годин: аудиторних – 1,5 самостійної роботи – 1,7	Освітньо-кваліфікаційний рівень: доктор філософії	28 год
		Практичні
		28 год
		Самостійна робота
		64 год
		Вид контролю: іспит

2. Мета, завдання та очікувані результати навчальної дисципліни.

Метою навчальної дисципліни «Технології створення програмних продуктів. Компонентне програмування» є вивчення основних концептуальних положень сучасних технологій програмування й створення програмних продуктів, інструментальних засобів розробки програм у рамках цього напрямку та формування у свідомості аспірантів необхідних понять про компонентно-орієнтовану парадигму програмування.

Завдання навчальної дисципліни:

- ознайомлення та оволодіння сучасними методами програмної інженерії;
- оволодіння сучасними національними та міжнародними стандартами в галузі розробки програмного забезпечення;
- отримання необхідних навичок з використання переваг компонентно-орієнтованої парадигми програмування.

Очікувані результати навчальної дисципліни

Загальні компетентності:

- ЗК1. Аналіз і синтез. Здатність до аналізу та синтезу на основі логічних аргументів та перевірених фактів; навички управління інформацією (уміння знаходити та аналізувати інформацію з різних джерел).
- ЗК2. Дослідницька здатність. Компетентності ініціювати та виконувати (індивідуально або в науковій групі) наукові дослідження, що приводять до отримання нових знань; здатність застосовувати знання на практиці.
- ЗК3. Гнучкість мислення. Набуття гнучкого способу мислення, який дає можливість зрозуміти й розв'язати проблеми та задачі, зберігаючи при цьому критичне відношення до усталених наукових концепцій.
- ЗК4. Креативність. Потенціал креативності у генеруванні ідей та досягненні наукових цілей.
- ЗК5. Групова робота. Здатність до взаємодії (робота в команді); здатність працювати в міждисциплінарній команді.
- ЗК7. Міжнародний кругозір. Здатність працювати в міжнародному середовищі, ставитися із повагою до національних та культурних традицій, способів роботи інших членів групи.
- ЗК8. Популяризаційні навички. Вміння спілкуватися із нефахівцями; здатність провести усну презентацію та написати зрозумілу статтю за результатами проведених досліджень, а також щодо сучасних концепцій у прикладній математиці для нефахівців у цій галузі; здатність спілкуватися з експертами з інших галузей.
- ЗК11. Етичні установки. Етика поведінки в наукових дослідженнях; турбота про якість результату.

- ЗК12. Особисті якості. Здатність до навчання, саморозвитку та самовдосконалення протягом життя; визначеність і наполегливість щодо поставлених завдань і взятих обов'язків.

Фахові компетентності:

- ФК2. Здатність застосовувати сучасну методологію, загальні та спеціальні методи наукового дослідження за спеціальністю та обраною спеціалізацією.
- ФК 5. Уміння будувати адекватні математичні моделі, досліджувати побудовані моделі та визначати можливості їх застосування.
- ФК 8. Здатність здійснювати дослідження прикладних математичних задач з використанням спеціальних математичних пакетів та аналізувати отримані дані.
- ФК18. Здатність ефективно аналізувати великі набори різномірних даних із використанням методів машинного навчання, статистики, штучного інтелекту тощо.
- ФК19. Знання сучасних технологій створення та захисту програмних продуктів, компетентність у алгоритмізації обчислювальних процесів та методів, знання та уміння використовувати сучасні мови програмування

Програмні результати навчання:

- ПРН 1. Знати та розуміти традиційні та передові тенденції в галузі прикладної математики, етапи історичного розвитку математичних знань і парадигм, формування поглибленої системи наукових знань за напрямом досліджень.
- ПРН 3. Знати основні принципи, форми та методи проведення наукових досліджень в галузі прикладної математики, володіти математичним апаратом в об'ємі, достатньому для професійної діяльності.
- ПРН 4. Вміти будувати та досліджувати математичні моделі фізичних, соціально-економічних, технічних, біологічних тощо процесів, систем, процесів, формалізувати проблеми, описані природною мовою, за допомогою математичних методів, застосовувати загальні підходи до математичного моделювання конкретних процесів.
- ПРН5. Мати навички використання загальних офісних та спеціалізованих програмних засобів за напрямом професійної діяльності, вміти користуватися спеціалізованими інтернет-ресурсами.
- ПРН9. Вміти організувати неперервний власний саморозвиток і самовдосконалення, набуття гнучкого способу мислення при творчому дослідженні, в тому числі прикладних математичних проблем.
- ПРН16. Вміти застосовувати сучасні технології алгоритмізації обчислювальних процесів, методів програмування, програмної реалізації чисельних і символьних алгоритмів.

За результатами вивчення навчальної дисципліни здобувач доктора філософії повинен:

знати:

- базові етапи процесу розробки програмного забезпечення;
- методи виміру показників якості програмних систем;
- основні положення стандартів ДСТУ 2844-94, ДСТУ 2941-94, ДСТУ EN ISO 9001:2015 та ДСТУ ISO/IEC/IEEE 15289:2019;
- основи програмування на мові Компонентний Паскаль у середовищі BlackBox Component Builder (інтерфейс, методи введення та виведення даних, поняття компіляції та вивантаження модуля, структуру модуля, відмінності від Паскаля).

вміти:

- обґрунтовано вибирати відповідні методи розробки програмних продуктів;
- робити тестування, налагодження та документування розроблених програмних продуктів;
- використовувати переваги компонентного підходу у проектуванні програмних комплексів та систем, розробляти програми на мові Компонентний Паскаль у середовищі BlackBox Component Builder.

3. Програма навчальної дисципліни

Тема 1. Основні напрямки програмної інженерії: науковий, інженерний, управлінський, виробничий. [2:11-49; 3:305-349; 10:17-71]

1. Програмна інженерія як наукова дисципліна.
2. Програмна інженерія як інженерна дисципліна.
3. Програмна інженерія як управлінська та виробнича дисципліна.
4. Ядро знань SWEBOOK.

Тема 2. Моделі життєвого циклу програмного забезпечення. [2:50-64; 5:24-43; 8:15-28; Д2:42-54]

1. Основні процеси життєвого циклу програмного забезпечення.
2. Каскадна та інкрементна моделі життєвого циклу програмного забезпечення.
3. Спиральна та еволюційна моделі життєвого циклу програмного забезпечення.

Тема 3. Сучасні методології проектування програмного забезпечення. [1:37-126; 2:90-97; 5:44-68]

1. Модульне програмування.
2. Структурне програмування: моделі SADT, SSADM, IDEF0.
3. Об'єктно-орієнтоване програмування. Мова UML.
4. Аспектно-орієнтоване програмування.
5. Генеруюче (породжуюче) програмування.
6. Агентне програмування.
7. Алгебраїчне та експлікативне програмування.

Тема 4. Моделі якості та надійності програмного забезпечення. [1:289-308, 328-353; 2:252-281; 3:272-278; 4:334-354; 6:79-84; 8:79-84; 10:339-407, 700-729; Д2:76-82]

1. Моделі якості програмних систем.
2. Базові поняття моделей надійності програмних систем. Класифікація Хетча.
3. Марковські та пуассонівські моделі надійності.

Тема 5. Верифікація та тестування програмного забезпечення. [1:221-273; 2:155-181; 3:282-291; 4:305-333; 6:84-94; 8:85-95; 10:226-254; Д2:65-69; Д4: 235-256]

1. Методи доведення правильності програм.
2. Верифікація та валідація програм.
3. Статичні та динамічні методи тестування програмних систем.
4. Класифікація помилок. Методи пошуку помилок.

Тема 6. Супровід програмного забезпечення. [10:255-282; Д1; Д2:69-75]

1. Основні етапи супроводу програмного забезпечення.
2. Використання зворотнього проектування, реінженірінгу та рефракторингу для супроводу програмного забезпечення.
3. Метрики оцінки якості супроводу програмного забезпечення.

Тема 7. Нормативно-методична підтримка життєвого циклу програмного забезпечення. [5:69-90; Д5; Д6; Д7; Д8; Д9]

1. Класифікація стандартів проектування інформаційних систем.
2. Основні положення стандартів ДСТУ 2844-94, ДСТУ 2941-94, ДСТУ EN ISO 9000:2015, ДСТУ EN ISO 9001:2015.
3. Основні положення стандарту ДСТУ ISO/IEC/IEEE 15289:2019.

Тема 8. Сучасні методи керування програмним проектом. [1:365-396; 2:282-301; 3:268-272; 6:29-40; 10:641-699]

1. Основні поняття та задачі менеджменту проекту.
2. Методи CPM (Critical Path Method) та PERT (Program Evaluation and Review Technique).
3. Методи керування ризиками у проекті.

Тема 9. Ідеологія компонентно-орієнтованого програмування. [2:111-116; 3:122-132; 10:464-472; Д2:40-42]

1. Поняття компоненту. Відмінність компонентів від об'єктів. Поняття збірки.
2. Підходи до моделювання варіацій компонентного підходу в сучасній практиці проектування та реалізації програмних комплексів та систем.
3. Моделі для компонентно-орієнтованої розробки програмного забезпечення.

Тема 10. Основні особливості компонентного програмування. [1:71-89; 3:165-184; 7; 10:473-489]

1. Життєві цикли та середовища компонентів. Переваги компонентного програмування.
2. Здатність компоненту до опису власних властивостей та вмінь. Взаємодія та незалежність компонентів.
3. Компонентна алгебра. Відношення між компонентами.

Тема 11. Основи програмування на мові Компонентний Паскаль у середовищі BlackBox Component Builder. [9:1-74, 87-112, 201-238, 251-292, 313-348, 381-398; Д3:7-11, 156-193]

1. Проект Оберон. Історія виникнення мови Компонентний Паскаль та середовища BlackBox.
2. Інтерфейс середовища BlackBox. Створення найпростіших програм у Компонентному Паскалі.
3. BlackBox/Компонентний Паскаль: типи даних, властивості модуля StdLog, форматування тексту програм.
4. BlackBox/Компонентний Паскаль: компіляція та налагодження програм.
5. BlackBox/Компонентний Паскаль: умовний оператор та оператор вибору. Цикли.
6. BlackBox/Компонентний Паскаль: комбінований тип даних. Робота з масивами.
7. BlackBox/Компонентний Паскаль: процедури та методи.
8. BlackBox/Компонентний Паскаль: робота з файлами та потоками.

4. Структура навчальної дисципліни

Назва теми	Кількість годин			
	Усього	у тому числі		
		Лекції	Практичні заняття	Самостійна робота
1	2	3	4	5
Тема 1. Основні напрямки програмної інженерії: науковий, інженерний, управлінський, виробничий.	9	2	2	5
Тема 2. Моделі життєвого циклу програмного забезпечення.	9	2	2	5
Тема 3. Сучасні методології проектування програмного забезпечення.	18	4	4	10
Тема 4. Моделі якості та надійності програмного забезпечення.	9	2	2	5
Тема 5. Верифікація та тестування програмного забезпечення.	9	2	2	5
Тема 6. Супровід програмного забезпечення.	9	2	2	5
Тема 7. Нормативно-методична підтримка життєвого циклу програмного забезпечення.	9	2	2	5
Тема 8. Сучасні методи керування програмним проектом.	9	2	2	5
Тема 9. Ідеологія компонентно-орієнтованого програмування.	9	2	2	5
Тема 10. Основні особливості компонентного програмування.	9	2	2	5
Тема 11. Основи програмування на мові Компонентний Паскаль у середовищі BlackBox Component Builder.	21	6	6	9
Усього годин	120	28	28	64

5. Теми практичних занять

№ з/п	Назва теми	Кількість годин
1	Основні напрямки програмної інженерії: науковий, інженерний, управлінський, виробничий.	2
2	Моделі життєвого циклу програмного забезпечення.	2
3	Сучасні методології проектування програмного забезпечення.	4
4	Моделі якості та надійності програмного забезпечення.	2
5	Верифікація та тестування програмного забезпечення.	2
6	Супровід програмного забезпечення.	2
7	Нормативно-методична підтримка життєвого циклу програмного	2

№ з/п	Назва теми	Кількість годин
	забезпечення.	
8	Сучасні методи керування програмним проектом.	2
9	Ідеологія компонентно-орієнтованого програмування.	2
10	Основні особливості компонентного програмування.	2
11	Основи програмування на мові Компонентний Паскаль у середовищі BlackBox Component Builder.	6
	Разом	28

6. Темати для самостійної роботи

№ з/п	Назва теми	Кількість годин
1	Основні напрямки програмної інженерії: науковий, інженерний, управлінський, виробничий.	5
2	Моделі життєвого циклу програмного забезпечення.	5
3	Сучасні методології проектування програмного забезпечення.	10
4	Моделі якості та надійності програмного забезпечення.	5
5	Верифікація та тестування програмного забезпечення.	5
6	Супровід програмного забезпечення.	5
7	Нормативно-методична підтримка життєвого циклу програмного забезпечення.	5
8	Сучасні методи керування програмним проектом.	5
9	Ідеологія компонентно-орієнтованого програмування.	5
10	Основні особливості компонентного програмування.	5
11	Основи програмування на мові Компонентний Паскаль у середовищі BlackBox Component Builder.	9
	Разом	64

7. Методи навчання

При вивченні дисципліни використовуються:

1. Дидактичні методи – лекції з використанням мультимедійних презентацій.
2. Практичні методи: практичні роботи з використанням прикладного програмного забезпечення
3. Метод самостійного навчання.
4. Активні методи: експрес опитування.
5. Словесні методи навчання: лекції, бесіди

8. Методи контролю

Використовуються наступні методи контролю:

1. Поточний контроль знань:
 - усне опитування;
 - виконання практичних робіт;
 - виконання дослідницьких завдань.
2. Підсумковий контроль знань (іспит):
 - екзаменаційні білети.

9. Критерії оцінювання результатів навчання

Розподіл балів, які отримує аспірант

Аудиторна робота											Самостійна робота	Підсумковий контроль (іспит)	Загальна кількість отриманих балів
теми													
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11			
3	3	4	6	4	4	4	4	4	4	10	10	40	100

Шкала оцінювання

Сума балів за всі види навчальної діяльності	Оцінка за національною шкалою
	для іспиту
90-100	«відмінно» - здобувач повно і всебічно розкриває питання теми, вільно оперує поняттями і термінологією, демонструє глибокі знання джерел, має власну точку зору стосовно відповідної теми і може аргументовано її доводити
75-89	«добре» - загалом рівень знань здобувачів відповідає викладеному вище, але мають місце деякі упущення при виконанні завдань, обґрунтування неточні, не підтверджуються достатньо обґрунтованими доказами
60-74	«задовільно» - здобувач розкрив питання, в загальних рисах, розуміє їх суть, намагається робити висновки, але при цьому припускається грубих помилок, матеріал викладає нелогічно і не самостійно
35-59	«незадовільно» - з можливістю повторного складання, здобувач не в змозі дати відповідь на поставлене запитання або відповідь неправильна, не розуміє суті питання, не може зробити висновки
0-34	«незадовільно» - з обов'язковим повторним вивченням дисципліни

10. Орієнтовний перелік питань до підсумкового контролю (іспиту)

1. Програмна інженерія як наукова дисципліна.
2. Програмна інженерія як інженерна дисципліна.
3. Програмна інженерія як управлінська та виробнича дисципліна.
4. Ядро знань SWEBOK.
5. Основні процеси життєвого циклу програмного забезпечення.
6. Каскадна та інкрементна моделі життєвого циклу програмного забезпечення.
7. Спиральна та еволюційна моделі життєвого циклу програмного забезпечення.
8. Модульне програмування.
9. Структурне програмування: моделі SADT, SSADM, IDEF0.
10. Об'єктно-орієнтоване програмування. Мова UML.
11. Аспектно-орієнтоване програмування.
12. Генеруюче (породжуюче) програмування.
13. Агентне програмування.
14. Алгебраїчне та експлікативне програмування.
15. Моделі якості програмних систем.
16. Базові поняття моделей надійності програмних систем. Класифікація Хетча.
17. Марковські та пуассонівські моделі надійності програмних систем.
18. Методи доведення правильності програм.
19. Верифікація та валідація програм.
20. Статичні та динамічні методи тестування програмних систем.
21. Класифікація помилок. Методи пошуку помилок.
22. Основні етапи супроводу програмного забезпечення.
23. Використання зворотнього проектування, реінженірінгу та рефакторингу для супроводу програмного забезпечення.
24. Метрики оцінки якості супроводу програмного забезпечення.
25. Класифікація стандартів проектування інформаційних систем.
26. Основні положення стандартів ДСТУ 2844-94, ДСТУ 2941-94, ДСТУ EN ISO 9000:2015, ДСТУ EN ISO 9001:2015.
27. Основні положення стандарту ДСТУ ISO/IEC/IEEE 15289:2019.
28. Основні поняття та задачі менеджменту проекту.
29. Методи CPM (Critical Path Method) та PERT (Program Evaluation and Review Technique).
30. Методи керування ризиками у проекті.
31. Поняття компоненту. Відмінність компонентів від об'єктів. Поняття збірки.
32. Підходи до моделювання варіацій компонентного підходу в сучасній практиці проектування та реалізації програмних комплексів та систем.
33. Моделі для компонентно-орієнтованої розробки програмного забезпечення.

34. Життєві цикли та середовища компонентів. Переваги компонентного програмування.
35. Здатність компоненту до опису власних властивостей та вмінь. Взаємодія та незалежність компонентів.
36. Компонентна алгебра. Відношення між компонентами.
37. Проект Оберон. Історія виникнення мови Компонентний Паскаль та середовища BlackBox.
38. Інтерфейс середовища BlackBox. Створення найпростіших програм у Компонентному Паскалі.
39. BlackBox/Компонентний Паскаль: типи даних, властивості модуля StdLog, форматування тексту програм.
40. BlackBox/Компонентний Паскаль: компіляція та налагодження програм.
41. BlackBox/Компонентний Паскаль: умовний оператор та оператор вибору. Цикли.
42. BlackBox/Компонентний Паскаль: комбінований тип даних. Робота з масивами.
43. BlackBox/Компонентний Паскаль: процедури та методи.
44. BlackBox/Компонентний Паскаль: робота з файлами та потоками.

11. Рекомендована література

Основна література дисципліни

1. Лаврищева Е.М. Методы программирования : теория, инженерия, практика. Київ : Наукова думка, 2006. 453 с.
2. Лаврищева К.М. Програмна інженерія : підручник. Київ : Академперіодика, 2008. 319 с.
3. Лаврищева Е.М., Грищенко В.Н. Сборочное программирование. Основы индустрии программных продуктов : 2-е изд. Київ : Наукова думка, 2009. 372 с.
4. Галіцин В.К., Сидоренко Ю.Т., Потапенко С.Д. Технологія програмування і створення програмних продуктів : навчальний посібник. Київ : КНЕУ, 2009. 372 с.
5. Карпенко М.Ю., Манакова Н.О., Гавриленко І.О. Технології створення програмних продуктів та інформаційних систем : навчальний посібник. Харків : ХНУМГ ім. О.М. Бекетова, 2017. 93 с.
6. І.Л. Бородкіна, Г.О. Бородкін. Інженерія програмного забезпечення. Київ : НУБіП України, 2018. 251 с.
7. Супруненко О.О., Гребенович Ю.Є. Інструментальні засоби компонентно-орієнтованого моделювання програмних систем та PN-патерни. Черкаси : Видавець Чабаненко Ю.А., 2022. 82 с.
8. Рудий Т.В., Паранчук Я.С., Сенік В.В. Алгоритмізація та програмування. Частина 1. Структурне програмування : навчальний посібник. Львів : ЛьвДУВС, 2023. 240 с.

9. J.S. Warford. Computing Fundamentals: The Theory and Practice of Software Design with BlackBox Component Builder. Vieweg & Sohn, 2002. 611p.
10. I. Sommerville. Software Engineering, 10th edition. Pearson, 2016. 810 p.

Допоміжна література дисципліни

1. Коцовський В.М. Супровід програмних систем : методичний посібник. Ужгород: "Говерла", 2016. 52 с.
2. Дегтярьова Л.М., Гроза П.М., Сомов С.В. Технології розробки програмного забезпечення : навчальний посібник. Полтава : ПолтНТУ, 2017. 218 с.
3. N. Wirth, J. Gutknecht. Project Oberon. The Design of an Operating System and Compiler (electronic edition, 2005).
4. N. Loubser. Software Engineering for Absolute Beginners: Your Guide to Creating Software Products. APress, 2021. 334p.
5. ДСТУ ISO/IEC/IEEE 15289:2019. Інженерія систем і програмних засобів. Уміст інформаційних об'єктів життєвого циклу (документації) (ISO/IEC/IEEE 15289:2019, IDT). [Чинний від 2019-08-01]. Вид. офіц. Київ, 2019. 76 с.
6. ДСТУ ISO 9000:2015. Системи управління якістю. Основні положення та словник термінів (ISO 9000:2015, IDT). [Чинний від 2015-12-21]. Вид. офіц. Київ : ДП «УкрНДНЦ», 2016. 49 с.
7. ДСТУ ISO 9001:2015. Системи управління якістю. Вимоги (ISO 9001:2015, IDT). [Чинний від 2015-12-31]. Вид. офіц. Київ : ДП «УкрНДНЦ», 2016. 30 с.
8. ДСТУ 2844-94. Програмні засоби ЕОМ. Забезпечення якості. Терміни та визначення. [Чинний від 1996-01-01]. Вид. офіц. Київ : Держстандарт України, 1995. 19 с.
9. ДСТУ 2941-94. Системи оброблення інформації. Розроблення систем. Терміни та визначення. [Чинний від 1996-01-01]. Вид. офіц. Київ : Держстандарт України, 1995. 13 с.