

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДНІПРОВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ОЛЕСЯ ГОНЧАРА

ЗАТВЕРДЖУЮ

Ректор

 Сергій ОКОВИТИЙ

« 28 » листопада 2026 р.



ПОГОДЖЕНО

Проректор з наукової роботи

 Олег МАРЕНКОВ

« 28 » листопада 2026 р.

ПРОГРАМА

ДОДАТКОВОГО ВСТУПНОГО ВИПРОБУВАННЯ

для здобуття ступеня доктора філософії

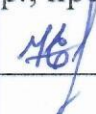
за спеціальністю F2 «Інженерія програмного забезпечення»

освітньо-наукова програма Інженерія програмного забезпечення

на основі освітнього ступеня магістра (освітньо-кваліфікаційного рівня спеціаліста)
з іншої галузі знань (спеціальності)



Розглянуто на засіданні вченої ради факультету
прикладної математики та інформаційних технологій
від «16» квітня 2026 р.; протокол № 8

Голова вченої ради  (Олена КІСЕЛЬОВА)

Дніпро-2026

Програма додаткового вступного іспиту для конкурсного відбору вступників до аспірантури для здобуття ступеня вищої освіти доктора філософії (PhD) за спеціальністю **F2 Інженерія програмного забезпечення** освітньо-наукова програма Інженерія програмного забезпечення (на основі освітнього ступеня магістра (освітньо-кваліфікаційного ступеня спеціаліста)) –Д: ДНУ, 2026. - 10с.

Розробники:

1. Байбуз О.Г., доктор технічних наук, професор, гарант освітньої програми, керівник проектної групи, завідувач кафедри інженерії програмного забезпечення та інформаційних технологій.
2. Мацуга О.М., кандидат технічних наук, доцент кафедри інженерії програмного забезпечення та інформаційних технологій.
3. Божуха Л.М., кандидат фізико-математичних наук, доцент кафедри інженерії програмного забезпечення та інформаційних технологій.
4. Ємельяненко Т.Г., кандидат технічних наук, доцент кафедри інженерії програмного забезпечення та інформаційних технологій.

Програма додаткового вступного іспиту ухвалена:

- на засіданні кафедри інженерії програмного забезпечення та інформаційних технологій (протокол № 9 від 14 квітня 2026 р.)

Завідувач кафедри інженерії програмного забезпечення та інформаційних технологій



(Олег БАЙБУЗ)

1. ЗМІСТ РОЗДІЛІВ ТА ТЕМ ДИСЦИПЛІН

Розділ 1. АЛГОРИТМИ ТА ОБЧИСЛЮВАЛЬНА СКЛАДНІСТЬ

1.1. Основи структури даних і алгоритми

1.1.1. Поняття алгоритму. Визначення його часової та просторової (за обсягом пам'яті) складності

1.1.2. Поняття абстрактного типу даних. Абстрактні типи даних: стеки, списки, вектори, словники, множини, мультимножини, черги, черги з пріоритетами

1.1.3. КORTEЖІ, множини, словники, одно- та двобічнозв'язні списки. Реалізація абстрактних типів даних з оцінюванням складності операцій

1.1.4. Базові алгоритми та їх складність: пошук, сортування (прості сортування вибором, вставками, обмінами та удосконалені сортування деревом, сортування Шелла, швидке сортування)

1.1.5. Алгоритми на графах та їх складність: пошук в ширину і глибину; пошук зв'язних компонентів; побудова кістякового дерева; побудова найкоротших шляхів з виділеної вершини; побудова найкоротших шляхів між двома вершинами

1.2. Стратегії розроблення алгоритмів

1.2.1. Стратегія «розділяй та володарюй» та приклади застосування.

1.2.2. Стратегія балансування та приклади застосування.

1.2.3. Динамічне програмування та приклади застосування.

1.2.4. Оцінювання складності алгоритму під час застосування кожної стратегії

1.3. Моделі обчислень

1.3.1. Імперативний та декларативний підходи до програмування.

1.3.2. Розв'язні, напіврозв'язні та нерозв'язні проблеми. Проблема зупинки.

Розділ 2. АРХІТЕКТУРА ОБЧИСЛЮВАЛЬНИХ СИСТЕМ

2.1. Функції бінарної логіки

2.2. Подання даних на рівні машин

2.2.1. Позиційні системи числення. Двійкова, вісімкова, шістнадцяткова системи числення. Беззнаковий код цілих чисел. Доповнювальний код цілих чисел. Основні арифметичні операції над цілими числами в беззнаковому та доповнювальному кодах.

2.2.2. Принципи зображення дійсних чисел в пам'яті програми у форматі з плаваючою комою. Переваги та недоліки форматів чисел з плаваючою комою. Основні арифметичні операції над дійсними числами у форматі з плаваючою комою та їхні проблеми.

2.3. Пристрої введення-виведення. Поняття шини комп'ютера

2.4. Функціональна організація обчислювальних систем

2.4.1. Структура комп'ютера, класична архітектура фон Неймана, гарвардська архітектура.

2.4.2. Ієрархічний принцип побудови пам'яті - регістрова, кеш, оперативна пам'ять, зовнішня пам'ять. CPU.

2.4.3. Периферійні пристрої

Тема 3. БАЗИ ТА СХОВИЩА ДАНИХ

3.1. Ключі та нормалізація даних: основні нормальні форми (1NF, 2NF, 3NF, BCNF)

3.2. Основні концепції систем баз даних: модель даних; мова запитів; транзакція; ACID-властивості транзакції, індексування; резервне копіювання та відновлення; розподіленість і реплікація даних; безпека даних

3.3. Моделювання даних: створення моделі даних для інформаційної системи; концептуальна, логічна, фізична моделі даних; ER-модель; нотації ER-моделей

3.4. Реляційні бази даних: особливості організації та зберігання даних у реляційних базах даних; основні характеристики реляційних баз даних; DBMS (Database Management System)

3.5. Побудова запиту: мови SQL (structured query language), DDL (Data Definition Language), DML (Data Manipulation Language), DCL (Data Control Language), TCL (Transaction Control Language)

3.6. Обробка запитів: основні операції реляційної алгебри: відбір (selection), проєкція (projection), об'єднання (union), перетин (intersection), різниця (difference), декартовий добуток (cartesian product), об'єднання за атрибутом (Join), ділення (Division)

Тема 4. ІНЖЕНЕРІЯ СИСТЕМ І ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1. Складні та великі системи

4.1.1. Властивості систем: емерджентність, адитивність, еквівінальність

4.1.2. Відкриті та закриті системи; класифікація за призначенням, походженням, видом елементів, способом організації

4.1.3. Спільне та відмінності складних і великих систем

4.2. Моделі систем. Склад і структура системи; моделі типу чорної та білої скриньки. Концептуальні, математичні і комп'ютерні моделі. Зв'язок між системою та моделлю; ізо- та гомоморфізм

4.3. Інформаційні системи

4.3.1. Поняття, цілі, значення, класифікація за функціональністю, масштабом, сферою застосування

4.3.2. Забезпечення інформаційних систем: організаційне, інформаційне, математичне, програмне, технічне, лінгвістичне, методичне, правове

4.4. Аналіз вимог

4.4.1. Класифікація вимог до програмного забезпечення, джерела та методи збирання вимог

4.4.2. Вимоги користувача (варіанти використання та історії користувачів)

4.4.3. Функціональні та нефункціональні вимоги, обмеження; структуризація функціональних вимог.

4.5. Проектування програмного забезпечення

4.5.1. Види проектування: Структурне проектування (Structural Design)

Об'єктно-орієнтоване проектування (Object-Oriented Design) Функціональне

проектування (Functional Design) Архітектурне проектування (Architectural Design) Інтерфейсне проектування (Interface Design)

4.5.2. Парадигми проектування: функціональна декомпозиція згори донизу, архітектура, орієнтована на дані, об'єктно-орієнтований аналіз та проектування, подієво-керована архітектура.

4.5.3. Ідентифікація класів предметної області. UML-діаграми ієрархії класів: моделювання підсистем, класів та зв'язків між ними.

4.5.4. Проектування сценаріїв реалізації варіантів використання на основі UML-діаграм послідовностей та комунікації.

4.5.5. Основні патерни проектування: MVC, Abstract Factory, Facade, Decorator, Flyweight, Visitor, Observer, Proxy, Strategy, Chain of Responsibility.

4.6. Реалізація програмного забезпечення

4.6.1. Вимоги до оформлення коду: стиль, розбиття на структуровані одиниці, найменування змінних, класів, об'єктів.

4.6.2. Засоби автоматичної генерації програмного коду

4.6.3. Налаштування: точки зупинки (Breakpoints), спостереження за змінними (Variable Watch), виведення на консоль (Console Output), налагоджувач (Debugger), аналізатори коду (Code Analyzers).

4.6.4. Керування конфігурацією та версіями програмного забезпечення.

4.6.5. Постійна інтеграція/постійне впровадження (Continuous Integration/Continuous Delivery).

4.7. Забезпечення якості: спільне та відмінності процесів тестування, верифікації, валідації

4.7.1. Тестування методами білої та чорної скрині.

4.7.2. Рівні тестування: модульний, інтеграційний, системний, валідаційний.

4.7.3. Розробка через тестування (Test-driven development).

4.7.4. Додаткові техніки верифікації та валідації: інспекція коду, перевірка на відповідність стандартам і вимогам, оцінювання зручності використання та користувацького досвіду, перевірка продуктивності та масштабованості.

4.8. Командна робота, підходи до розробки програмного забезпечення (ПЗ)

4.8.1. Класичні моделі розробки ПЗ: каскадна (водоспадна), ітераційна, інкрементна.

4.8.2. Промислові технології розробки ПЗ: RUP, MSF, Agile, Scrum, Extreme Programming (XP), Kanban.

4.8.3. Ролі та обов'язки у команді проекту, переваги командної роботи, ризики та складність такої співпраці.

4.8.4. Основні етапи планування і виконання ІТ проекту. Життєвий цикл ІТ проекту.

Тема 5. ОСНОВИ МОВ ПРОГРАМУВАННЯ

5.1. Сутність і види мов програмування

5.1.1. Поняття класу та об'єкта в об'єктно-орієнтованому програмуванні; конструктор і деструктор, інтерфейс і реалізація.

- 5.1.2. Базові концепції об'єктно-орієнтованого програмування: абстракція, інкапсуляція, спадкування, поліморфізм.
- 5.1.3. Зв'язки між класами в об'єктно-орієнтованому програмуванні: асоціація, агрегація, композиція, спадкування, залежність, реалізація.
- 5.1.4. Порівняння процедурного та об'єктно-орієнтованого програмування.
- 5.2. Принципи та сфера застосування видів програмування: функціональне, логічне, подійно-орієнтоване, реактивне, узагальнене програмування
- 5.3. Моделі паралельних обчислень: класифікація Флінна
- 5.4. Трансляція та виконання: компілятор, інтерпретатор, компоновальник

3. ОСНОВНА ЛІТЕРАТУРА

Розділ 1. Алгоритми та обчислювальна складність

1. Методичні рекомендації до вивчення розділу «Програмування складних динамічних структур даних» // Укладачі: д-р техн. наук, проф. В. В. Гнатушенко, канд. техн. наук, старш. викл. А. О. Єгоров, старш. викл. Н. О. Соколова, РВВ ДНУ, 2014.
<http://repository.dnu.dp.ua:1100/upload/fb08aa20d29ac072a18e7be785dd91bbProgramming.pdf>
2. Спірінцева О. В., Литвинов О. А., Герасимов В. В. Java-технології та мобільні пристрої. Алгоритми і структури даних: навчальний посібник. Дніпропетровськ : Вид-во ДНУ ім. О. Гончара, 2016. 140 с.
3. Кублій Л. І. Алгоритми та структури даних. Основи алгоритмізації : підручник / Л. І. Кублій. - Київ: КПІ ім. І. Сікорського, 2022. - 528 с.

Розділ 2. Архітектура обчислювальних систем

1. Комп'ютерні мережі. [Книга 2. Архітектура комп'ютерів] : навчальний посібник / С. П. Євсєєв, Н. В. Дженюк, М. Ю. Толкачов та ін. – Харків, – Львів : «Новий Світ – 2000», 2025. – 346 с. (Укр. мов.)
2. Тарарака В.Д. Архітектура комп'ютерних систем: навчальний посібник. – Житомир : ЖДТУ, 2018. – 383 с.
3. John L. H., D. A. Patterson. Computer Architecture: A Quantitative Approach. – Elsevier Science & Technology. – 2017 – 492 p.
4. Tanenbaum A.S., Austin T. Structured Computer Organization. – Prentice Hall, 2012. – 800 p.

Розділ 3. Бази та сховища даних

1. Навчально-методичний посібник до курсового проектування з дисципліни «Організація баз даних» / Л. Г. Ахметшина, А. О. Єгоров, В. В. Герасимов. — Дніпро, 2018. — 68 с.
2. Берко А. Ю., Верес О. М., Пасічник В. В. Організація баз даних та знань. — Магнолія 2006. — 2024. — 435 с.

3. Connolly T., Begg C. Database Systems: A Practical Approach to Design, Implementation, and Management, Global Edition. — Longman (Pearson Education), 2014. — 1442 с.

Розділ 4. Інженерія систем і ПЗ

1. Certified Tester. Foundation Level Syllabus / K. Olsen, T. Parveen, R. Black, D. Friedenberg, M. Hamburg, J. McKay, M. Posthuma, H. Schaefer, R. Smilgin, M. Smith, S. Toms, S. Ulrich, M. Walsh, E. Zakaria. — International Software Testing Qualifications Board, 2019. — 93 p.
2. Database Systems: Design, Implementation, & Management/Coronel C., Morris S. — Видавництво: Cengage Learning, 2019. — 837с.
3. Database System Concepts/Silberschatz A., Korth H.F., Sudarshan S. — Видавництво: McGraw-Hill Education, 2019. — 1373с.
4. Мартін Р. Чиста архітектура: мистецтво розробки програмного забезпечення. — Фабула, 2019. — 416 с.
5. Кормен Т. Вступ до алгоритмів: Пер. з англ. третього видання: [укр.] = Introduction to Algorithms / Т. Кормен, Ч. Лейзерсон, Р. Рівест, К. Стайн. — К.: К.І.С., 2019. — 1288 с.

Розділ 5. Основи мов програмування

1. Томас Г. Кормен, Чарлз Е. Лейзерсон, Роналд Л. Рівест, Кліффорд Стайн. Вступ до алгоритмів - К.І.С., 2023. - 1288 с.
2. Рудий Т. В. Алгоритмізація та програмування. Частина 1. Структурне програмування : навчальний посібник / Рудий Т. В., Паранчук Я. С., Сенік В. В. - Львів: Львівський державний університет внутрішніх справ, 2023. - 240 с.
3. Рудий Т. Алгоритмізація та програмування. Частина 2. Модульне програмування : навчальний посібник / Тарас Рудий, Ярослав Паранчук, Володимир Сенік. - Львів: Львівський державний університет внутрішніх справ, 2024. - 176 с.
4. Васильєв О. Програмування на C++ в прикладах і задачах: Навч. посіб. / О. Васильєв — Київ: Видавництво Ліра-К, 2017. — 382 с.

4. СТРУКТУРА БІЛЕТУ

Кожний білет єдиного фахового іспиту містить **25 тестових завдань**. Питання складено у формі обрання однієї вірної відповіді з чотирьох запропонованих.

Оцінка за відповідь на кожне питання може набувати одного з двох значень:

максимального значення 4 балів у випадку вірної відповіді;

мінімального значення 0 балів у випадку невірної відповіді.

Розподіл питань у білеті:

— за формою завдань:

	Кількість тестових завдань у варіанті	Кількість балів за тестове одне завдання	Максимальна кількість балів
Розділ 1. Алгоритми та обчислювальна складність	5	4	20
Розділ 2. Архітектура обчислювальних систем	5	4	20
Розділ 3. Бази та сховища даних	5	4	20
Розділ 4. Інженерія систем і ПЗ	5	4	20
Розділ 5. Основи мов програмування	5	4	20
Загалом	25	4	100

5. КРИТЕРІЇ ОЦІНКИ ВІДПОВІДЕЙ

Екзаменаційний білет складається з 25 тестових питань. Кожне завдання екзаменаційного білету оцінюється у чотири бали.

Остаточна оцінка за екзаменаційну роботу вступника виставляється як сума отриманих балів по кожному питанню екзаменаційного білета. Максимально можлива оцінка – 200 балів.

Таблиця переведення тестових балів фахового іспиту до шкали 100–200

Тестовий бал	Бал за шкалою 100-200
0-12	100
16	116
20	120
24	124
28	128
32	132
36	136
40	140

Тестовий бал	Бал за шкалою 100-200
44	144
48	148
52	152
56	156
60	160
64	164
68	168
72	172

Тестовий бал	Бал за шкалою 100-200
76	176
80	180
84	184
88	188
92	192
96	196
100	200