

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

«ЗАТВЕРДЖУЮ»

президент Національного університету  
«Києво-Могилянська академія»



Сергій КВІТ

«18» травня 2026 р.

## ПРОГРАМА ФАХОВОГО ІСПИТУ

для здобуття ступеня магістра за спеціальністю  
**F2 Інженерія програмного забезпечення**  
(галузь знань: **F «Інформаційні технології»**;  
освітньо-наукова програма: «Інженерія програмного забезпечення»)

Схвалено  
Вченою радою  
факультету інформатики  
(протокол № 7 від 14 травня 2026 р.)

## 1. ЗАГАЛЬНІ ПОЛОЖЕННЯ

Фаховий іспит за спеціальністю **F2 Інженерія програмного забезпечення** передбачений «Правилами прийому на навчання до Національного університету «Києво-Могилянська академія» в 2026 році» для тих вступників, які вступають для здобуття ступеня магістра за спеціальними умовами вступу відповідно до п. 84 розділу V Правил прийому.

Фаховий іспит має за мету з'ясування рівня професійних компетенцій, теоретичних знань і практичних навичок вступників зі спеціальності **F2 Інженерія програмного забезпечення**, визначення їхньої готовності до засвоєння відповідної освітньо-наукової програми.

Фаховий іспит за спеціальністю **F2 Інженерія програмного забезпечення** проводиться очно в комп'ютерних класах НаУКМА із застосуванням технологій комп'ютерного тестування на платформі DistEdu із відеофіксацією процесу. Результати тестування зберігаються в корпоративному хмарному сховищі та/або системі DistEdu.

НаУКМА забезпечує відеозапис вступного іспиту зі спеціальності не менше ніж з двох відеокамер, розміщення на сайті університету, внесення посилання на цей запис в ЄДЕБО впродовж трьох робочих днів після оприлюднення оцінок вступників, а також зберігання відеоматеріалів та письмових робіт протягом одного року.

**Вступний тест містить 24 завдання.**

Завдання 1-20 – завдання з короткою відповіддю та вибором однієї або декількох правильних відповідей (множинний вибір). Завдання 21-24 – завдання відкритої форми з розгорнутою відповіддю.

**Тривалість фахового вступного випробування – 90 хв.**

## 2. ЗМІСТ ПРОГРАМИ

Вступне фахове випробування зі спеціальності F2 Інженерія програмного забезпечення до вступу на навчання за освітньо-кваліфікаційним рівнем магістр в 2026 році передбачає перевірку базового рівня знань з основних дисциплін, здобутих бакалаврами в результаті успішного засвоєння таких дисциплін:

1. Комп'ютерні алгоритми
2. Дискретна математика
3. Програмування
4. Системне програмування
5. Бази даних

### **Комп'ютерні алгоритми**

1. Системи числення.
2. Класичні алгоритмічні системи (Нормальні алгоритми Маркова, Машина Тюрінга, Алгоритмічна система Поста).
3. Складність алгоритмів. Задачі класу P і NP.
4. Алгоритми сортування. Реалізація, складність, особливості.
5. Комбінаторні алгоритми.

6. Лінійні списки, стеки та черги.
7. Графи. Задання графів у пам'яті. Обходи графів.
8. Деревя. Програмна реалізація задання дерев у пам'яті. Обходи дерев. Ізоморфізм дерев.
9. Хеш-таблиці реалізація, складність, особливості.
10. Неорієнтовані графи – внутрішня структура, використання, класичні алгоритми та задачі над неорієнтованими графами.
11. Орієнтовані графи – внутрішня структура, використання, класичні алгоритми та задачі над орієнтованими графами.
12. Жадібний метод. Основи методу. Алгоритм Дейкстри: пошук найкоротших шляхів для графу з одним джерелом. Коди Хаффмана.
13. Задачі оптимізації. Загальна характеристика.
14. Метод «розділяй і пануй». Загальна схема, приклади застосування. Узагальнене рівняння декомпозиції. Розв'язання рекурентних рівнянь
15. Бектрекінг. Загальна характеристика. Зворотна рекурсія. Задача про 8 ферзів. Сума підмножин. Гамільтонові цикли. Задача про рюкзак.
16. Динамічне програмування (ДП). Загальна характеристика методу. Одномірне динамічне програмування. Двовимірне динамічне програмування. Задача про зростаючу підпоследовність. Задача про паліндром. Задача обчислення добутку матриць. Мемоізація (заповнення кеша зверху-вниз) і табуляція на прикладі задачі знаходження чисел Фібоначчі. Техніки програмування у методі ДП. Спадне динамічне програмування. Порівняння ДП і методу розділяй і пануй. Аналіз застосовності та ефективності ДП.

### Дискретна математика

1. Послідовності, що задаються рекурентними співвідношеннями. Розв'язання рекурентностей типу Фібоначчі.
2. Основні поняття теорії множин – елемент, підмножина, універсальна множина, порожня множина, характеристична функція. Операції над множинами – об'єднання, перетин, різниця, симетрична різниця, доповнення. Основні властивості цих операцій. Узагальнені закони дистрибутивності та де Моргана.
3. Декартів добуток множин та його властивості, приклади, узагальнення. Множини  $B^A, 2^A$ .
4. Основні принципи комбінаторики. Задача про підрахунок кількості функцій, визначених на скінченних множинах, та кількості k-елементних розміщень на множині.
5. Комбінації без повторень. Основні властивості коефіцієнтів  $C_n^k$ . Біном Ньютона та наслідки з нього.

6. Перестановки з повтореннями (перестановки типів). Формула для кількості перестановок.
7. Поліноміальні коефіцієнти як коефіцієнти в розкладі полінома  $(x_1 + x_2 + \dots + x_k)^n$ .
8. Формули включень та виключень.
9. Формальні мови та граматики. Класифікація граматик за Хомським.
10. Регулярні граматики та скінченні автомати.
11. Машина Тюрінга.
12. Поняття n-арного відношення на множинах. Бінарні відношення. Відношення еквівалентності. Класи еквівалентності. Теорема про розбиття множини на класи еквівалентності. Поняття фактор-множини. Приклади.
13. Основні поняття теорії графів. Орієнтовані графи, неорієнтовані графи, прості графи. Суміжність, інцидентність вершин, ребер, степінь вершини, регулярні графи. Ізоморфізм графів. Приклади.
14. Поняття ланцюга, циклу. Ейлерові графи. Критерій наявності ейлерового циклу та напівейлерового ланцюга в графі. Ліс та дерева. Еквівалентність різних означень дерева.

## Програмування

### *Процедурне і об'єктно-орієнтоване програмування*

1. Особливості передачі параметрів та повернення значення функції. Указники і відсилки (pointers and references).
2. Створення і ініціалізація об'єктів, довизначення конструкторів, замовчуваний конструктор, обмеження прав доступу до конструктора.
3. Поверхнєве і глибоке копіювання об'єктів, ініціалізація і присвоєння, копіювальний конструктор.
4. Довизначення (overloading) арифметичних операцій, оператора присвоєння, оператора індексування.
5. Узагальнені функції (function template).
6. Параметризовані класи (class template).
7. Відкрите, закрите і захищене успадкування.
8. Успадкування із спільного базового класу. Домішки (mix-in).
9. Статичне і динамічне зв'язування: поліморфізм, віртуальні функції.
10. Успадкування інтерфейсу і успадкування реалізації. Абстрактні класи.

### *Функціональне програмування.*

#### 1. Списки

- Функції вищого порядку над списками: map, filter, foldl, foldr
- Арифметичні послідовності
- Формувачі списків
- Нескінченні списки

## 2. Функції

- Визначення функції і виконання функції
- Анонімні функції
- Умови (охоронні вирази)
- Конструкції let, where, case. Двовимірний синтаксис

## 3. Типи даних

- Базові типи даних. Кортежі і функції
- Синоніми типів (type)
- Створення нових типів (data)
- Співставлення зі зразком, види зразків

## 4. Класи типів

- Поняття класу типів. Екземпляри класу
- Клас типів Eq, Ord, Enum. Автоматичне визначення екземплярів класів типів

## 5. Модулі і дії введення-виведення.

- Означення модуля. Експорт і імпорт модулів
- Конфлікти імен
- Дії введення-виведення. Тип IO a.

## *Веб-програмування.*

1. Механізми клієнт-серверної взаємодії.
2. PHP як мова програмування. – Процедурні риси PHP
3. JavaScript як мова програмування. – Процедурні риси JS
4. JS на стороні клієнта.

## **Системне програмування.**

1. Визначення граматики. Класифікація граматик за Хомським.
2. Співставлення класів граматик з автоматами та розпізнавачами.
3. Регулярні мови і скінчені автомати. Побудова детермінованого автомату за недетермінованим.
4. Синтаксичний аналіз і дерева граматичного розбору. Класифікація методів організації синтаксичного розбору
5. Безперебірні методи синтаксичного аналізу. LL(k) граматики.
6. Багатопроцесорність. Симетричні та асиметричні системи. Системи з загальною пам'яттю.
7. Операційні системи. Ядро ОС.

## **Бази даних.**

1. Поняття інформаційної системи. Класифікація інформаційних систем. Поняття та властивості БД та СУБД.
2. Інформаційна модель концептуального рівня “сутність-зв’язок”.
3. Реляційна модель.
4. Алгоритм переходу від ER-моделі до реляційної моделі.
5. Цілісність даних. Підтримка цілісності в реляційній моделі.

6. Реляційна алгебра Кодда. Основні операції над реляціями.
7. Мова SQL: оператор вибірки даних SELECT.
8. Функціональні залежності. Аксиоми Армстронга. .
9. Властивості декомпозиції схем реляцій. Алгоритми перевірки на з'єднання без втрат та на збереження залежностей.

### 3. ЛІТЕРАТУРА ДЛЯ ПІДГОТОВКИ

1. Боднарчук Ю. В., Олійник Б. В. Основи дискретної математики: Навч. посіб. — К.: Вид. дім «Києво-Могилянська академія», 2009. — 159 с.
2. Оленко А. Я., Ядренко М. Й. Дискретна математика: навч.-метод. посіб. - К.: НаУКМА, 1996. — 83 с.
3. Дрінь С.С., Дяченко С.М., Захарійченко Ю.О., Пилявська О.С., Чорней Р.К. Конспект лекцій з вищої математики. – К.: Видавничий дім «АртЕк», 2014. – 118с. 17.
4. Глибовець М.М., Олецький О.В. Штучний інтелект, Видавничий дім “КМ Академія ” 2002.
5. Бублик В. В. Об'єктно-орієнтоване програмування : підручник для студентів, які навчаються за напрямками "Комп'ютерні науки", "Комп'ютерна інженерія", "Програмна інженерія", "Інформатика", "Прикладна математика" / В. В. Бублик. - Київ : ІТ-книга, 2015. - 637 с
6. Глибовець М.М. Основи комп'ютерних алгоритмів. - К.: Вид. дім. "КМ Академія", 2003. - 450с.
7. Кренивич А.П. Алгоритми і структури даних. Підручник. – К.: ВПЦ "Київський Університет", 2021. – 200 с.
8. Ковалюк Т.В. Алгоритмізація та програмування: Підручник. – Львів: «Магнолія 2006», 2013. – 400 с.
9. Бардус І.О. Бази даних у схемах: Навчальний посібник – Харків: «Дісаплюс», 2017, 133

### 4. КРИТЕРІЇ ОЦІНЮВАННЯ

Результати фахового вступного випробування визначаються шляхом переведення набраних вступником тестових балів у шкалу числового інтервалу [0; 200].

Повна відповідь на кожне з тестових питань 1-20 оцінюється в 3 бали, часткова відповідь (для тестових завдань з множинним вибором) – від 0,5 до 2,5 балів. Якщо відповідь неправильна або відсутня – 0 балів.

Відповіді на завдання 21-24 оцінюються за 10-бальною шкалою:

- 10 – завдання виконано бездоганно;
- 9 – завдання виконано з незначним зауваженням;
- 7 – завдання виконано з незначними зауваженнями;
- 5 – завдання виконано частково;
- 3 – завдання виконано частково і з суттєвими зауваженнями;
- 1 – невдала спроба виконати завдання;
- 0 – відповідь відсутня;

2,4,6,8 – проміжні оцінки між двома сусідніми балами.

Таким чином, максимально можлива кількість набраних тестових балів складає:

$$20 \cdot 3 + 4 \cdot 10 = 100.$$

Фахова комісія, провівши аналіз складності тесту за результатами набраних вступниками тестових балів, визначає пороговий бал “склав / не склав”  $T_{min}$ . Крім того, визначається максимальний тестовий бал  $T_{max}$ , отриманий серед вступників в результаті складання тесту.

Підсумковий результат  $R$  фахового випробування обчислюється за формулою:

$$R = 100 \cdot \frac{T - T_{min}}{T_{max} - T_{min}} + 100,$$

де  $T$  – набраний вступником тестовий бал.

Абітурієнт вважається таким, що склав фахове вступне випробування за спеціальністю F2 Інженерія програмного забезпечення, якщо його підсумкова оцінка за виконання тестових завдань становить **100 – 200 балів**.

У випадку, якщо результати фахового вступного випробування за 200-бальною шкалою становлять менше 100 балів, тобто тестовий бал менший від порогового балу “склав / не склав”, абітурієнт вибуває з конкурсного відбору на спеціальність F2 Інженерія програмного забезпечення.

**Голова фахової  
атестаційної комісії**

**Алла НАГІРНА**