

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«Київський політехнічний інститут імені Ігоря Сікорського»



**Затверджую**

Голова Приймальної комісії  
Ректор

Анатолій МЕЛЬНИЧЕНКО

28.03.2025 р.

дата

**Факультет прикладної математики**  
*повна назва факультету/навчально-наукового інституту*

**ПРОГРАМА**  
**фахового іспиту**

для вступу на освітньо-наукову та освітньо-професійну програми підготовки магістра  
«Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем»


**за спеціальністю F 2 Інженерія програмного забезпечення**

Програму ухвалено:

Вченою Радою факультету прикладної математики

Протокол № 8 від 24 лютого 2025 р.

Голова Вченої Ради

 Іван ДИЧКА

## ***ВСТУП***

Фаховий іспит проводиться зі вступниками на навчання за освітньою програмою підготовки магістрів “Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем” за спеціальністю F2 Інженерія програмного забезпечення.

Метою Програми комплексного фахового випробування (далі — Програми) є надання вступникам вичерпної інформації про склад, структуру, критерії оцінювання результатів комплексного фахового випробування для вступу на освітньо-наукову та освітньо-професійну програми підготовки магістрів за спеціальністю F2 Інженерія програмного забезпечення (освітня програма «Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем»).

Програма фахового іспиту передбачає перевірку набуття вступником компетентностей та результатів навчання, що визначені стандартом вищої освіти за спеціальністю F2 Інженерія програмного забезпечення для першого (бакалаврського) рівня вищої освіти.

Також в програмі наведений приклад екзаменаційного білету, який складається з трьох теоретичних запитань.

Фаховий іспит проводиться в очній або дистанційній формі з використанням технологій дистанційного навчання та сервісу відеотелефонного зв'язку (Zoom, GoogleMeet тощо) із обов'язковою відеофіксацією процесу проведення іспиту.

Порядок проведення екзамену відповідає «Положенню про поточний, календарний та семестровий контроль результатів навчання в КПІ ім. Ігоря Сікорського» (<https://osvita.kpi.ua/node/32>) та «Регламентам проведення семестрового контролю та захистів кваліфікаційних робіт та атестаційних екзаменів в дистанційному режимі» (<https://osvita.kpi.ua/node/148>).

## ОСНОВНИЙ ВИКЛАД

1. Ключі та нормалізація даних: основні нормальні форми (1NF, 2NF, 3NF, BCNF, 4NF).
2. Основні концепції систем баз даних: модель даних; мова запитів; транзакція; ACID-властивості транзакції, індексування; резервне копіювання та відновлення; розподіленість та реплікація даних; безпека даних.
3. Моделювання даних: створення моделі даних для інформаційної системи; концептуальна, логічна, фізична моделі даних; ER-модель; нотації ER-моделей.
4. Реляційні бази даних: особливості організації та зберігання даних у реляційних базах даних; основні характеристики реляційних баз даних; DBMS (Database Management System).
5. Побудова запиту: мови SQL (structured query language), DDL (Data Definition Language), DML (Data Manipulation Language), DCL (Data Control Language), TCL (Transaction Control Language).
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).
7. Розподілені бази даних/хмарні обчислення: доступність, масштабованість, виклики, технології.
8. Особливості, переваги і недоліки моделей напівструктурованих і неструктурованих баз даних: моделі даних Ключ-значення (Key-Value), Документо-орієнтовані (Document-Oriented), Стовпцево-орієнтовані (Column-Family), Графові (Graph), Масив-орієнтовані (Array-Based).
9. Поняття класу та об'єкта; конструктор і деструктор, інтерфейс та реалізація
10. Базові концепції ООП: абстракція, інкапсуляція, спадкування, поліморфізм
11. Зв'язки між класами: асоціація, агрегація, композиція, спадкування, залежність, реалізація
12. Поняття алгоритму. Визначення його часової та просторової складності
13. Поняття абстрактного типу даних.
14. Стандартні абстрактні типи даних: стеки, списки, вектори, словники, множини, мультимножини, черги, черги з пріоритетами
15. Кортежі, множини, словники, одно- та двозв'язні списки.
16. Реалізація абстрактних типів даних з оцінюванням складності операцій
17. Базові алгоритми та їх складність: пошук, сортування (прості сортування видаленням, вставками, обмінами та удосконалені сортування деревом, сортування Шела, швидке сортування)
18. 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
19. Джерела та методи збирання вимог
20. Вимоги користувача: модель вимог на основі прецедентів (варіантів використання) (Use Case Diagram), історії користувачів (user story). Вимоги до описів варіантів використання
21. Класифікація вимог до програмного забезпечення: функціональні та нефункціональні вимоги, обмеження, структуризація функціональних вимог.
22. Моделювання проєкту з UML: діаграми статичні та динамічні, логічні та фізичні

23. Види проектування: архітектурне (верхній рівень) та деталізоване проектування (класів, атрибутів, операцій), проектування інтерфейсу користувача
24. Парадигми проектування: функціональна декомпозиція згори вниз, архітектура, орієнтована на дані, об'єктно-орієнтований аналіз та проектування, подієво-керована архітектура
25. Ідентифікація класів предметної області. UML-діаграми ієрархії класів: моделювання підсистем, класів та зв'язків між ними
26. Проектування сценаріїв реалізації варіантів використання на основі UML-діаграм послідовностей та комунікації
27. Роль архітектури. Стандартні архітектури: клієнт-серверна та n-рівнева архітектура, Model View Controller
28. Архітектурні моделі та патерни проектування (Abstract Factory, Facade, Decorator, Flyweight, Visitor, Observer, Proxy, Strategy, Chain of Responsibility)
29. Вимоги до оформлення коду: стиль, розбиття на структуровані одиниці, найменування змінних, класів, об'єктів тощо
30. Засоби автоматичної генерації програмного коду на основі трансформацій UML-модель - код ООП-мовою, повторне використання коду ПЗ
31. Налаштування: Точки зупинки (Breakpoints), Спостереження за змінними (Variable Watch), Виведення на консоль (Console Output), Налаштовувач (Debugger), Аналізатори коду (Code Analyzers)
32. Керування конфігурацією програмного забезпечення та контроль версій
33. Постійна інтеграція/постійне впровадження (Continuous Integration/Continuous Delivery)
34. Призначення, спільне та відмінності процесів тестування, верифікації, валідації
35. Види тестів: модульні, інтеграційні, регресійні, системні, валідаційні
36. Тестування методами білої та чорної скрині
37. Розробка через тестування (Test-driven development)
38. Додаткові техніки верифікації та валідації: інспекція коду, перевірка на відповідність стандартам і вимогам, оцінювання зручності використання та користувацького досвіду, перевірка продуктивності та масштабованості
39. Класичні моделі розробки ПЗ: каскадно-водопадна, ітераційна, інкрементна
40. Промислові технології розробки. RUP- уніфікований підхід, керований варіантами використання, архітектурно-центрований, ітераційний, інкрементний
41. Ролі та обов'язки у програмній команді, переваги командної роботи, ризики та складність такої співпраці
42. Технології гнучкої розробки ПЗ та їх особливості: Agile, Scrum, Extreme Programming (XP), Kanban
43. Моделі керування командною роботою (на основі UML діаграм Ганта і Перта)

## **КРИТЕРІЇ ОЦІНЮВАННЯ ФАХОВОГО ВИПРОБУВАННЯ**

Критерії оцінювання відповіді студента враховують повноту та правильність відповіді, а також здатність студента узагальнювати отримані знання, застосовувати загальні та специфічні наукові методи, принципи та закони на конкретних прикладах; аналізувати, інтерпретувати та оцінювати отримані результати.

Екзаменаційний білет містить три питання:

1. «Бази та сховища даних» - практично-орієнтоване завдання
2. «Об'єктно-орієнтоване програмування» або «Основи структури даних і алгоритми» - теоретично-орієнтоване завдання
3. «Інженерія систем і програмного забезпечення» - теоретично-орієнтоване завдання.

Відповідь на перше (практично-орієнтоване) завдання оцінюється за 40-бальною шкалою. Критерії оцінювання наведені у табл. 1. Друге та третє (теоретично-орієнтовані) завдання білета оцінюються за 30-бальною шкалою. Критерії оцінювання наведені у табл. 2.

Таблиця 1

Бали	Характеристика відповіді
37-40	Повна ґрунтовна відповідь на запитання з відповідними поясненнями.
31-36	Правильна відповідь, але хід міркувань наведений не повністю або відсутня одна позиція відповіді.
21-30	Правильна відповідь, але з деякими неточностями, або неповна (освітлено не менше двох третин питання).
11-20	Відповідь не зовсім точна або неповна (освітлено не менше половини питання).
1-10	Відповідь приблизна, містить неточності чи некоректності, або неповна (не менше третини інформації).
0	Відповідь здебільшого невірна, містить некоректності. Наведена невелика частина відповіді (не менше чверті інформації).

Критерії оцінювання теоретично-орієнтованого завдання

Бали	Характеристика відповіді
27-30	Повна ґрунтовна відповідь на запитання з відповідними поясненнями.
23-26	Правильна відповідь, але хід міркувань наведений не повністю або відсутня одна позиція відповіді.
18-22	Правильна відповідь, але з деякими неточностями, або неповна (освітлено не менше двох третин питання).
14-17	Відповідь не зовсім точна або неповна (освітлено не менше половини питання).
10-13	Відповідь приблизна, містить неточності чи некоректності, або неповна (не менше третини інформації).
5-9	Відповідь здебільшого невірна, містить некоректності. Наведена невелика частина відповіді (не менше чверті інформації).
1-4	Завдання почали виконувати, але не наведено коректних правильних відповідей.
0	Відповідь на питання відсутня.

Бали за всі три завдання білету підсумовуються. **Максимальна можлива кількість балів:  $40 + 30 + 30 = 100$  балів.**

Якщо вступник на комплексному фаховому випробуванні отримав оцінку **нижчу за 60 балів** або не з'явився на випробування без поважної причини, то вважається, що він не склав вступне випробування, і до подальшої участі в конкурсі він не допускається.

Вступники, які користувалися на екзамені недозволеними допоміжними матеріалами, пристроями, або працювали не самостійно видаляються із екзамену і отримують оцінку **Незадовільно**.

Перескладання комплексного фахового випробування з метою підвищення оцінки не дозволяється.

При обчисленні конкурсного балу застосовується шкала оцінювання 100. „200 балів. Перерахунок оцінки рейтингової системи оцінювання в шкалу 100. „200 балів наведено у Таблиці 3.

Таблиця відповідності оцінок PCO (60...100 балів)  
оцінкам 200-бальної шкали (100...200 балів)

шкала PCO	шкала 100...200	шкала PCO	шкала 100...200	шкала PCO	шкала 100...200	шкала PCO	шкала 100...200
60	100	70	140	80	160	90	180
61	105	71	142	81	162	91	182
62	110	72	144	82	164	92	184
63	115	73	146	83	166	93	186
64	120	74	148	84	168	94	188
65	125	75	150	85	170	95	190
66	128	76	152	86	172	96	192
67	131	77	154	87	174	97	194
68	134	78	156	88	176	98	196
69	137	79	158	89	178	99	198
						100	200

## ПРИКЛАД ТИПОВОГО ЗАВДАННЯ КОМПЛЕКСНОГО ФАХОВОГО ВИПРОБУВАННЯ

Спеціальність F2 Інженерія програмного забезпечення

Освітня програма Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем

### ФАХОВИЙ ІСПИТ ДЛЯ ВСТУПУ ДО МАГІСТРАТУРИ ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № \_

1. У базі даних фірми створюється таблиця клієнтів фірми, яка складається з наступних атрибутів:
  - id клієнта (clientjd),
  - ім'я (name),
  - номер телефону (phone\_number),
  - додаткова інформація для нотаток (additionalinformation).

Напишіть SQL-код створення даної таблиці у системі управління базами даних PostgreSQL з визначенням типів даних та обмежень, що відповідають зазначеним атрибутам, якщо за одним номером телефону може бути закріплений один клієнт. Поясніть обрані типи даних та обмеження.

2. Поняття абстрактного типу даних.
3. Керування конфігурацією програмного забезпечення та контроль версій.

## **I. Рекомендована література з теми «Бази та сховища даних»**

1. Павловський В.І., Петрашенко А.В., Победа Д.В. Бази даних та засоби управління. Практикум: навч. посібник. Київ : КПІ ім. Ігоря Сікорського, 2022. 112 с. URL: <https://ela.kpi.ua/handle/123456789/46193>
2. Гайна Г.А. Основи проектування баз даних: Навчальний посібник. Київ : Кондор, 2018. 204 с.
3. Пасічник В.В., Резніченко В.А. Організація баз даних та знань. Київ: Вид. група ВНУ, 2006. 384 с.
4. Гайдаржи В.І., Ізварін І.В. Бази даних в інформаційних системах. Київ: Університет «Україна», 2018. 418 с.
5. Берко А.Ю., Верес О.М., Пасічник В.В. Системи баз даних та знань : підручник для студентів вищих начальних закладів. Львів : Видавництво "Магнолія 2006", 2021 - 2 книги.
6. P. Rob, C. Coronel. Database Systems: Design, Implementation, and Management. Course Technology, 2011. 728 p.
7. C. J. Date. Introduction to Database Systems. Addison-Wesley Longman, Inc., 2004. 983 p.
8. H. Garcia-Molina, J. Ullman, J. Widom. Database Systems: The Complete Book. Pearson, 2008. 1248 p.
9. PostgreSQL: Documentation. PostgreSQL: The world's most advanced open source database. URL: <https://www.postgresql.org/docs/>
10. Regina O. Obe, Leo S. Hsu. PostgreSQL: Up and Running: A Practical Guide to the Advanced Open Source Database. O'Reilly Media, 2017. 483 p.

## **II. Рекомендована література з теми «Об'єктно-орієнтоване програмування»**

1. Об'єктно-орієнтоване програмування [Текст]: метод. вказівки до викон. курсової роботи для студ. напряму підготовки 6.050103 «Програмна інженерія» / Уклад.: Т.М. Заболотня. К.: ІВЦ "Видавництво «Політехніка»", 2011. - 80 с.
2. Шаблони проектування: методичні вказівки до виконання лабораторних робіт з дисципліни «Об'єктно-орієнтоване програмування» для студентів напряму підготовки 6.050103 «Програмна інженерія» [Електронне видання] / Т.М. Заболотня. К.: НТУУ «КПІ», 2015. - 154с.
3. Perkins B., Hammer J.V., Reid J.D. Beginning C# 7 Programming with Visual Studio 2017. John Wiley & Sons, Inc, 2018, 912pp. ISBN: 978-1-119-45868-5
4. M. Olsson. C# 7 Quick Syntax Reference: A Pocket Guide to the Language, APIs and Library, 2018, 180pp. ISBN-13 (pbk): 978-1-4842-3816-5

5. Bishop J. C# 3.0 Design Patterns, O'Reilly, 2008, 290pp. ISBN 10: 0-596-52773-X, ISBN 13: 978-0-596-52773-0
6. Будаї А. Дизайн-патерни - просто, як двері [Електронне видання] / А.Будаї. - Режим доступу: <http://designpattens.andriybuday.com/>
1. Алгоритми та структури даних: Навч. посібник для студентів тех. спец. / О.М. Кривонос, С.С. Жуковський. - Житомир : Вид-во ЖДУ ім. І. Франка, 2020. - 66 с.
2. Алгоритми та структури даних: Навч. посібник для студентів тех. спец. д. та з. ф-ми навч. / Мелешко Є.В., Якименко М.С., Поліщук Л.І. - Кропивницький, 2019. - 156 с.
3. Алгоритми і структури даних. Підручник / Крєневич А.П. - Київ : ВПЦ "Київський Університет", 2021. - 200 с.
4. Mehlhorn K. Data structures and algorithms 1: Sorting and searching. Vol. 1. Springer Science & Business Media, 2013.
5. Lee K.D., Hubbard S. Data Structures and Algorithms with Python. Springer, 2015.
6. Mueller J. P., Massaron L. Algorithms for Dummies. John Wiley & Sons, 2017.
7. Erickson J. Algorithms. 2019.
8. Wengrow J. A Common-Sense Guide to Data Structures and Algorithms. Pragmatic Bookshelf, 2020.
9. Bhasin H. Algorithms: Design and Analysis. Oxford University Press, 2015.
10. Roughgarden T. Algorithms Illuminated. SoundLikeYourself Publishing, 2017.
11. Cormen T. H., Leiserson Ch. E., Rivest R. L., Stein C. Introduction to Algorithms. MIT Press, 2009.
12. Stephens R. Essential Algorithms: a Practical Approach to Computer Algorithms using Python and C#. John Wiley & Sons, 2019.

### **III. Рекомендована література з дисципліни «Інженерія систем і програмного забезпечення»**

1. Дистанційний курс «Компоненти програмної інженерії - 1». КПІ ім. Ігоря Сікорського. Код доступу c5mtych (classroom.google.com)
2. Стеценко І.В. Моделювання систем: навч. посіб. [Електронний ресурс, текст] / І.В. Стеценко. - Черкаси: ЧДТУ, 2010. - 399 с. - ISBN 978-966-402-073-9
3. ISO 9126:1991, Інформаційна технологія. Оцінка програмного продукту. Характеристики якості та посібник щодо їх застосування. - Режим доступу: <https://studfile.net/preview/5300003/page:4/>
4. Томашевський В.М. Моделювання систем / В.М. Томашевськей - К.: Видавнича група BVH, 2005. - 352 с. - Режим доступу:

- [http://www.immsp.kiev.ua/postgraduate/Biblioteka\\_trudy/Tomashevsky\\_Model.system\\_2005.pdf](http://www.immsp.kiev.ua/postgraduate/Biblioteka_trudy/Tomashevsky_Model.system_2005.pdf)
5. Bourque P., Fairley R. SWEBOK V3.0. Guide to the Software Engineering Body of Knowledge. IEEE Computer Society. - Режим доступу: <https://cs.fit.edu/~kgallagher/Schtick/Serious/SWEBOKv3.pdf>
  6. ISO/IEC 12207:2008, System and software engineering. - Режим доступу: [http://www.ing.unisannio.it/cimitile/ingsw/dispense/ISO\\_12207.pdf](http://www.ing.unisannio.it/cimitile/ingsw/dispense/ISO_12207.pdf)
  7. Wiegers K. Software Requirements (Developer Best Practices). Microsoft, 2004. - 575 p.
  8. A Guide to the Business Analysis Body of Knowledge (BABOK Guide). IIBA, 2015. - 502 p. - ISBN-13 97978-1-927584-03-3.

**РОЗРОБНИКИ**

доцент кафедри ПЗКС



Тетяна ЗАБОЛОТНЯ

доцент кафедри ПЗКС



Леся ЛЮШЕНКО

доцент кафедри ПЗКС



Інна САЯПНА

Програму рекомендовано кафедрою програмного забезпечення комп'ютерних систем  
Протокол № 9 від «20» лютого 2025 р.

Завідувач кафедри



Євгенія СУЛЕМА