

**МОН УКРАЇНИ
НАЦІОНАЛЬНИЙ ТРАНСПОРТНИЙ УНІВЕРСИТЕТ**

ЗАТВЕРДЖУЮ



Проректор з навчальної роботи

Олександр ГРИЩУК

04 _____ 2022 р.

**ПРОГРАМА
АТЕСТАЦІЙНОГО ЕКЗАМЕНУ
за освітньо-професійною програмою
«ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»**

**першого (бакалаврського) рівня вищої освіти
за спеціальністю 121 «Інженерія програмного забезпечення»
галузі знань 12 «Інформаційні технології»**

**Освітня кваліфікація
Бакалавр з інженерії програмного забезпечення**

Київ – 2022

Програму атестаційного екзамену для атестації випускників першого (бакалаврського) рівня вищої освіти освітньо-професійної програми «Інженерія програмного забезпечення» за спеціальністю 121 «Інженерія програмного забезпечення» галузі знань 12 «Інформаційні технології» у 2021-2022 навчальному році розроблено кафедрою інформаційних систем і технологій.

Розглянуто та схвалено на засіданні кафедри інформаційних систем і технологій.

Протокол № 8 від 29 березня 2022 року.

Розглянуто та схвалено на засіданні Вченої Ради факультету транспортних та інформаційних технологій.

Протокол № 8 від 26 квітня 2022 року.

Розглянуто та схвалено на засіданні Науково-методичної ради Національного транспортного університету.

Протокол № 29 від 29 квітня 2022 року.

ЗМІСТ

Загальні положення.....	4
1. Дисципліна «Алгоритми та структури даних».....	5
2. Дисципліна «Конструювання програмного забезпечення».....	7
3. Дисципліна «Моделювання та аналіз програмного забезпечення».....	10
Критерії оцінювання досягнення результатів навчання	14
Додаток А. Форма білета атестаційного екзамену	17

ЗАГАЛЬНІ ПОЛОЖЕННЯ

Атестаційний екзамєн є формою атестації випускників першого (бакалаврського) рівня вищої освіти освітньо-професійної програми «Інженерія програмного забезпечення» за спеціальністю 121 «Інженерія програмного забезпечення» галузі знань 12 «Інформаційні технології» у 2021-2022 навчальному році. Атестація завершується видачею документа встановленого зразка про присудження здобувачу освітньої кваліфікації «Бакалавр з інженерії програмного забезпечення».

Атестаційний екзамєн передбачає оцінювання досягнення результатів навчання, визначених освітньо-професійною програмою «Інженерія програмного забезпечення» для атестації.

Програма атестаційного екзамєну для випускників першого (бакалаврського) рівня вищої освіти освітньо-професійної програми «Інженерія програмного забезпечення» за спеціальністю 121 «Інженерія програмного забезпечення» галузі знань 12 «Інформаційні технології» розроблена кафедрою інформаційних систем і технологій на основі цієї освітньо-професійної програми.

Атестаційний екзамєн випробування проводиться у письмовій формі з використанням тестових технологій. Процедура проведення атестаційного екзамєну може змінюватись у разі несприятливої безпекової ситуації.

Білет атестаційного екзамєну містить 14 запитань двох рівнів складності з основних профільюючих дисциплін.

Запитання першого рівня складності (з 1-го по 10-е запитання білета атестаційного екзамєну) передбачають вибір студентом відповіді із наведених у білеті 3 варіантів відповіді, з яких тільки один правильний.

Запитання другого рівня складності (з 11-го по 14-е запитання білета атестаційного екзамєну) передбачають надання студентом розгорнутої теоретичної відповіді.

Правильний на думку студента варіант відповіді на запитання першого рівня складності студент позначає безпосередньо на бланку білета атестаційного екзамєну.

Відповідь на запитання другого рівня складності студент наводить на аркушах для письмової відповіді.

1. ДИСЦИПЛІНА «АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ»

1. Основні поняття теорії алгоритмів

Поняття алгоритму і його основні характеристики. Вимоги до алгоритмів. Способи задання алгоритмів. Базові структури алгоритмів. Алгоритмічні моделі. Методи розробки алгоритмів.

2. Аналіз алгоритмів

Математичні основи аналізу алгоритмів. Ефективність алгоритмів. Характеристики продуктивності алгоритмів. О-нотація. Приклади аналізу алгоритмів. Часова та просторова складності алгоритмів. Асимптотична складність алгоритмів.

3. Формалізація поняття алгоритму

Машина Тьюрінга. Основні елементи машини Тьюрінга. Функціонування машини Тьюрінга. Нормальні алгоритми Маркова. Основні поняття та визначення нормальних алгоритмів Маркова. Структура та побудова нормальних алгоритмів Маркова. Основні поняття та визначення рекурсивних функцій.

4. Алгоритми пошуку і сортування

Алгоритми послідовного та бінарного пошуку. Основні алгоритми сортування: сортування вставками, вибором, обмінами (бульбашками) та їх порівняльний аналіз.

5. Алгоритми на графах

Пошук в глибину на графі. Пошук в ширину на графі. Знаходження найкоротших шляхів на графі.

6. Основи теорії автоматів

Основні поняття і властивості автоматів. Методи задання автоматів. Еквівалентні автомати. Мінімізація автоматів. Дві моделі скінчених автоматів: автомат Мілі та автомат Мура.

7. Структури даних

Основні поняття про структури даних. Структури даних: черга, зв'язаний список, хеш-таблиця, граф.

Орієнтовні запитання для атестаційного екзамену

Запитання першого рівня складності

1. Що таке алгоритм?
2. Як називається алгоритм, який для вирішення поставлених завдань використовує лише дії з числами?
3. Що таке область визначення алгоритму?
4. Що таке абстрактний алфавіт?
5. Що належать до основних вимог до алгоритму?
6. При якій умові алгоритм дає правильні результати лише для певних вхідних даних?
7. Що є базовими структурами алгоритмів?

8. Наведіть приклади основних типів формалізації алгоритмічних моделей.
9. Що називають часовою складністю алгоритму?
10. Що таке ємнісна складність алгоритму?
11. Що враховує асимптотична оцінка складності алгоритму?
12. Визначте асимптотичну складність функції.
13. Який тип складності має задана функція?
14. До якого типу належать швидкі алгоритми?
15. Яку асимптотичну складність має бінарний пошук?
16. Які процедури називаються рекурсивними процедурами?
17. Клас P?
18. За допомогою яких алгоритмів можна розв'язати задачі Клас NP?
19. Що передбачає бульбашкове сортування?
20. При якому типі сортування на кожному кроці алгоритму вибирається один з елементів вхідних даних і вставляється на потрібну позицію вже відсортованого масиву?
21. Що потрібно робити при побудові обходу графу в ширину?
22. Що потрібно використовувати при побудові обходу графу у глибину?
23. Який алгоритм передбачає знаходження найкоротших шляхів від однієї вершини до всіх інших у зваженому графі?
24. Що таке структури даних?
25. Що таке зв'язаний список?
26. Як називається структура, яка працює за принципом LIFO (Last In First Out – «останній прийшов – перший пішов»)?

Запитання другого рівня складності

1. Абстрактний алфавіт. Числові і логічні алгоритми.
2. Базові структури алгоритму. Навести приклади.
3. Способи задання алгоритмів. Структурні компоненти блок-схем.
4. Три основні типи формалізації алгоритмічних моделей.
5. Асимптотична складність алгоритму.
6. Типи складності алгоритмів – θ , O і Ω .
7. Класи складності алгоритмів P, NP.
8. Бульбашкове сортування.
9. Сортування вставками.
10. Сортування вибором.
11. Пошук в глибину на графі.
12. Пошук в ширину на графі.
13. Алгоритм Дейкстри знаходження найкоротших шляхів на графі.
14. Структура даних – черга.
15. Структура даних – зв'язаний список.
16. Структура даних – хеш-таблиця.
17. Структура даних – граф.

Список рекомендованої літератури

1. Клакович Л. М., Левицька С.М., Костів О.В. Теорія алгоритмів: Навч. посібник.— Львів: Видавничий центр ЛНУ імені Івана Франка, 2008. — 140с.
2. Коротєєва Т. О. Алгоритми та структури даних: навч. посіб. / Т. О. Коротєєва ; Нац. ун-т "Львів. політехніка". - Львів : Вид-во Львів. політехніки, 2014. - 279 с.
3. Глибовець М. М. Основи комп'ютерних алгоритмів [Текст] / М. М. Глибовець.— К. : Видавничий дім "КМ Академія", 2003. - 450 с.
4. Костів. О. В. Методи розробки алгоритмів: тексти лекцій / О. В. Костів, С. А. Ярошко ; Львівський національний ун-т ім. Івана Франка. – Л. : Видавничий центр ЛНУ імені Івана Франка, 2002. - 98 с.
5. Михальов О. І. Структури даних та алгоритми: навч. посіб. для студ. вищ. навч. закл. / Михальов, В. В. Крамаренко, К. М. Ялова, К. Ю. Новікова ; Дніпродзерж. держ. техн. ун-т. - Дніпродзержинськ : ДДТУ, 2010. – 284 с.

2. ДИСЦИПЛІНА «КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

1. Основи конструювання програмного забезпечення

Життєвий цикл програмного продукту. Коротка характеристика моделей життєвого циклу. Фундаментальні складові конструювання програмного забезпечення: мінімізація складності, очікування змін, конструювання з можливістю перевірки, стандарти у конструюванні. Управління процесом конструювання: моделі конструювання. Планування конструювання. Вимірювання в конструюванні.

2. Практичні аспекти конструювання

Проектування в конструюванні. Структурне проектування. Декомпозиція системи. Мови конструювання: конфігураційні мови, інструментальні мови й мови програмування. Кодування. Рефакторинг. Тестування в конструюванні. Статична перевірка коду. Специфікації. Повторне використання. Тестування Black box та glass box. Якість конструювання. “Bad smell” коди. Інтеграція.

3. Технології й інструменти конструювання програмного забезпечення

Середовища розробки програмного забезпечення. Основи Java для Android Studio, ML для отримання навичок вмотивовано обирати мову програмування для певних задач конструювання.

4. Системи керування версіями

Виникнення систем контролю версій. Перелік операцій, які повинні підтримуватися системою контролю версій. Автоматичне управління в системах контролю версій. Порівняння різниць, створення гілок, злиття,

вирішення конфліктів, маркування. Централізовані і розподілені системи контролю версій. Термінологія контролю версій. Особливості функціонування системи контролю версій. Постачальники Інтернет-хостингу для розробки програмного забезпечення та контролю версій. Розміщення репозиторію на GitHub. Особливості локальної роботи з розподіленою системою контролю версій Git. Створення файлів й розміщення їх у трьох локальних станах.

5. Створення мобільного застосунку

Складові інтерфейсу програмного середовища Android Studio (SDK, Emulator). Структура Android Manifest. Основи роботи з Activity. Android Context. Елементи інтерфейсу. Атрибути View та ViewGroup. Система зборки Gradle. Можливості Android Debug Bridge. Система контролю версій. Фрагменти й файли Preferences. Activity авторизації, Activity профіля. Додавання фрагментів. Toast. Додавання логіки авторизації. Створення персонажів. Налаштування та підключення програми для баз даних.

Орієнтовні запитання для атестаційного екзамену

Запитання першого рівня складності

1. До якого типу змінної належить BigInteger?
2. Що таке статично типізовані мови?
3. Що може зафіксувати динамічна перевірка коду?
4. Визначте, що з переліченого вважається небажаним під час кодування?
5. Що таке «чистий» код?
6. Коли виникає потреба у рефакторингу?
7. Коли рекомендують виконувати перевірку коду (code review)?
8. За допомогою чого необхідно контролювати доступ до несанкціонованої зміни глобальних даних (global data)?
9. Що таке "лінивий елемент"?
10. Коли рекомендується використовувати коментарі до коду?
11. Про що наступне формулювання "Система, що розробляється, повинна скоротити час на нарахування заробітної платні вдвічі"?
12. Про що наступне формулювання "Система, що розробляється, має надавати можливість вести облік працівників на підприємстві, передавати вибіркову інформацію про них різним рівням топ-менеджменту із високим рівнем надійності й безпеки"?
13. Про що наступне формулювання "Скановані копії документів працівника компанії можуть бути внесені, завантажені, відредаговані й видалені із системи, що розробляється"?
14. Що можна віднести до прикладів формулювання нефункціональних вимог?
15. Чим є GET <https://www.mysite.com/api/users/17>?
16. Що таке Facebook?

17. Що розуміють під "позолоченням " програмного продукту?
18. Що таке CASE-технології?
19. Що таке OpEx?
20. Що таке система контролю версій?
21. Для чого зазвичай створюються гілки в Git?
22. Що таке Tag в Git?

Запитання другого рівня складності

1. Сутність фундаментальних складових конструювання програмного забезпечення: мінімізація складності, очікування змін, конструювання з можливістю перевірки.
2. Статична перевірка коду. Статична типізація. Документування припущень.
3. Статична й динамічна перевірка коду.
4. Конструювання в різних моделях життєвого циклу розробки програмного забезпечення.
5. Принципи об'єднання окремих компонентів в єдину систему (інтеграція).
6. Технології й інструменти конструювання програмного забезпечення.
7. Середовища розробки програмного забезпечення.
8. Інструменти розробки кросплатформного програмного забезпечення.
9. Опис процесу планування конструювання.
10. Стандарти в конструюванні програмного забезпечення.
11. Системи контролю версій.
12. Управління версіями файлів в системі Git: особливості створення гілок і злиття.
13. Стратегії оптимізації коду.
14. Види і стратегії рефакторінгу.
15. Інструменти захисту програмного забезпечення від несанкціонованого використання й модифікації.
16. Інструменти захисту програмного забезпечення від неправильних вхідних даних.
17. Способи обробки помилок.
18. Основні типи даних. Створення власних типів даних. Нестандартні типи даних.
19. Налаштування: пошук і усунення дефектів у програмі. Інструменти.
20. Конструювання програмного забезпечення. Форматування і стиль.
21. Оцінка якості програмного продукту.
22. Функціональні й нефункціональні вимоги до програмного забезпечення.

23. Типи застосунків, особливості їх застосування й технології розробки.
24. Характеристики хмарних сервісів. Моделі фінансування.
25. Порівняння моделей хмарних сервісів: public, private, hybrid clouds. Хмарні платформи.
26. Порівняння типів хмарних сервісів: IaaS, PaaS, SaaS.

Список рекомендованої літератури

1. S. McConnell, Code Complete, 2nd ed., Microsoft Press, 2004. – 896 p.
2. B. Meyer, Object-Oriented Software Construction, second ed., Prentice Hall, 1997. – 1254 p.
3. Martin Fowler, Refactoring: Improving the Design of Existing Code. Publisher: Addison-Wesley Professional, 2nd edition, 2019. – 455 p.
4. Michael C. Feathers. Working Effectively with Legacy Code, Publisher: Pearson Education, Prentice-Hall – 2005.
5. Grant Allen. Android for Absolute Beginners. Getting Started with Mobile Apps Development Using the Android Java SDK. Apress Media, 2021. – 356 p.

3. ДИСЦИПЛІНА «МОДЕЛЮВАННЯ ТА АНАЛІЗ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

1. Основи моделювання програмного забезпечення

Основні принципи моделювання: декомпозиція, абстракція, узагальнення, демонстрація, використання формальних підходів. Властивості моделей. Метамодель. Синтаксична конструкція, семантика, практичні аспекти(прагматика). Передумови, постумови, інваріанти та обумовлене проектування. Мови моделювання програмного забезпечення: графічні й текстові. Мови специфікацій. Модель «концепція-знання» (C-K theory). Вступ до математичного моделювання та формальні нотації. Концептуальне моделювання. Побудова моделі опису процесів при структурному підході. Нотація IDEF0. Структурні карти Джексона, Константайна.

2. Види моделей

Моделі типу «чорної скриньки» і «скляної скриньки». Чотири основних представлення (views) як проекції властивостей моделі: структура, функції, поведінка, дані. Інформаційні моделі. Поведінкові моделі. Структурні моделі. Моделювання предметної області.

3. Моделювання програмного забезпечення із використанням уніфікованої мови моделювання uml

Призначення UML у розрізі проектування програмного забезпечення. Види діаграм. Діаграма варіантів використання/прецедентів (Use Case). Діаграма діяльності. Діаграма станів. Діаграма класів. Діаграма послідовності. Діаграма компонентів. Спрощена стратегія використання

UML-діаграм при моделюванні програмного забезпечення. Засоби розширення UML. Стереотипи. Профілі предметних областей. Помічені значення. Обмеження. Перехід від ділових моделей до моделей інформаційних систем.

4. Моделі (методології) розробки програмного забезпечення

Шаблон характеристик моделей розробки програмного забезпечення. Традиційні й гнучкі моделі. Предикативні й адаптивні моделі. Інкрементні й ітеративні моделі. Каскадна модель. V-модель. Сашімі модель. Інкрементна модель. Модель об'єднаного процесу (UP). Спіральна модель. Основні принципи і цінності гнучких моделей. Agile Manifesto. Scrum, Kanban, Lean Startup, DSDM, Extreme Programming (XP).

5. Моделювання бізнес процесів

Моделі динамічних дискретних систем подій. Теорія автоматів. Моделювання транспортних процесів за допомогою теорії мереж Петрі. Нотація бізнес-процесів Business Process Modeling Notation (BPMN).

6. Аналіз програмного забезпечення

Аналіз коректності (статичний аналіз програми, проведення модельних експериментів, перевірка моделі). Перевірка надійності програмного забезпечення. Аналіз видів і наслідків відмов (FMEA). Аналіз дерева відмов (FTA). Формальний аналіз коду. Автоматичне доведення теорем.

Орієнтовні запитання для атестаційного екзамену

Запитання першого рівня складності

1. Що таке структурні моделі?
2. Що таке Business Process Modeling Notation (BPMN)?
3. Що таке Мережі Петрі (Petri Nets)?
4. Що таке теорія «Концепція-Знання» (C-K theory)?
5. Які з тверджень характеризують прецеденти (варіанти використання)?
6. Що належить до поведінкових діаграм UML?
7. Що є результатом моделювання варіантів використання?
8. Який варіант використання коректний для системи «Інтернет-магазин книжок»?
9. Яке призначення UML?
10. Оберіть правильне твердження для UML.
11. Яка з властивостей є основною для моделі «Канбан»?
12. Яке твердження стосується Scrum?
13. Що з наступного справедливо щодо маніфесту Agile?
14. Що не належить до принципів моделювання програмних систем?
15. Які проблеми з моделлю водоспаду спонукали програмну індустрію створити гнучкі моделі розробки програмного забезпечення?
16. Чому під час проектування програмного забезпечення іноді важко визначити потреби та вимоги користувачів?

17. Які функції виконує Scrum-мастер?
18. Що належить до діаграми UML?
19. Що з наведеного справедливо для моделі водоспаду, V-моделі та моделі Сашімі?
20. Що з наведеного справедливо для адаптивної моделі розробки програмного забезпечення?
21. Виберіть етапи, що обов'язково виконується на кожному циклі спіральної моделі розробки програмного забезпечення.
22. За рахунок чого модель Сашімі допомагає скоротити тривалість проекту?
23. Що справедливо для моделі швидкого запуску (Lean Startup)?
24. Що називають моделлю системи типу «чорного ящика»?

Запитання другого рівня складності

1. Концептуальне моделювання програмного забезпечення.
2. Принципи моделювання програмного забезпечення. Властивості моделей.
3. Мови моделювання програмного забезпечення.
4. Характеристика різних типів моделей: інформаційні, поведінкові й структурні. Представлення моделей (Views).
5. Види діаграм UML. Стратегія використання UML-діаграм при моделюванні програмного забезпечення.
6. Засоби розширення UML для моделювання нових елементів, враховуючи специфіку предметної області.
7. Будівельні блоки UML. Правила й механізми UML.
8. Нотація UML. Види сутностей і зв'язків.
9. Моделювання структурних зв'язків, зв'язків залежності, наслідування узагальнення, асоціації, реалізації.
10. Характеристика структурних діаграм.
11. Характеристика діаграм поведінки.
12. Діаграми класів: моделювання логічної схеми бази даних.
13. Діаграми компонентів. Компоненти й інтерфейси.
14. Діаграми варіантів використання. Моделювання вимог до системи.
15. Діаграми взаємодії: моделювання потоків управління.
16. Діаграми діяльності: моделювання потоку робіт.
17. Діаграми станів: моделювання реактивних об'єктів.
18. Традиційні моделі розробки програмного забезпечення.
19. Гнучкі моделі розробки програмного забезпечення.
20. Ітеративні моделі розробки програмного забезпечення.
21. Адаптивні й предикативні моделі розробки програмного забезпечення
22. Характеристика спіральної моделі розробки програмного забезпечення.

23. Принципи й базові цінності маніфесту Agile.
24. Сутність й приклади використання моделі Lean.
25. Моделювання бізнес процесів. Приклади інструментів моделювання бізнес процесів.

Список рекомендованої літератури

1. D. Budgen. Software Design, 2nd ed., Addison-Wesley, 2003.
2. Sommerville, Software Engineering, 9th ed., Addison-Wesley, 2011.
3. S.J. Mellor and M.J. Balcer, Executable UML: A Foundation for Model-Driven Architecture, 1st ed., Addison-Wesley, 2002.
4. Петрик М.Р. Моделювання програмного забезпечення: науково-методичний посібник / М.Р. Петрик, О.Ю. Петрик – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2015. – 200 с.
5. Табунщик Г. В. Проектування та моделювання програмного забезпечення сучасних інформаційних систем / Г. В. Табунщик, Т.І. Каплієнко, О.А. Петрова – Запоріжжя : Дике Поле, 2016. – 250 с.

ЗАТВЕРДЖУЮ



Проректор з навчальної роботи

Олександр ГРИЩУК

29 04 2022 р.

КРИТЕРІЇ

**оцінювання досягнення результатів навчання
на атестаційному екзамені за освітньо-професійною програмою
«Інженерія програмного забезпечення»
першого (бакалаврського) рівня вищої освіти
за спеціальністю 121 «Інженерія програмного забезпечення»
галузі знань 12 «Інформаційні технології»**

Структура оцінки атестаційного екзамену

Оцінка *атестаційного екзамену* (за шкалою від 0 до 100 балів) складається із суми балів, виставлених атестаційною комісією в результаті перевірки письмової роботи, виконаної студентом під час атестаційного екзамену, за відповіді студента на кожне з 14 запитань білета атестаційного екзамену.

Порядок оцінювання досягнення результатів навчання

Оцінку атестаційного екзамену визначають у такому порядку:

- 1) виставляють бали за відповіді на кожне запитання білета атестаційного екзамену виходячи із наведених нижче критеріїв оцінювання відповідей;
- 2) обчислюють оцінку атестаційного екзамену за формулою:

$$O = \sum_{i=1}^{14} B_i,$$

де B_i – кількість балів за відповідь на i -е запитання.

Відповіді у чернетці не перевіряють та до уваги не беруть.

Критерії оцінювання відповідей на запитання

Відповідь на кожне запитання першого рівня складності (запитання з 1-го по 10-е, які передбачають вибір студентом відповіді із наведених у білеті 3 варіантів відповіді, із яких тільки один правильний) може бути оцінена у 2 бали (якщо вибрано правильну відповідь) або 0 балів (якщо вибрано

неправильну відповідь із запропонованих у білеті варіантів відповіді, або вибрано більше одного варіанта відповіді, або відповідь не надано).

Відповідь на кожне запитання другого рівня складності (запитання з 11-го по 14-е, які передбачають надання студентом розгорнутої теоретичної відповіді) може бути оцінена балами від 0 до 20.

Відповідь на запитання другого рівня складності оцінюють виходячи із наведених у таблиці характеристик відповіді.

Кількість балів	Характеристика відповіді
16–20	Повна, наведена у логічно правильній послідовності відповідь, яка свідчить про всебічні, систематизовані та глибокі знання матеріалу навчальної дисципліни; демонструє здатність студента вільно оперувати здобутими знаннями: диференціювати та інтегрувати їх, відтворювати та аналізувати отриману інформацію, робити обґрунтовані висновки та узагальнення, виявляти й відстоювати власну позицію, переконливо висловлювати думку та чітко формулювати відповідь. Як правило, таку оцінку отримує студент, який відповів на запитання не менше ніж на 90 %. Відповідь оцінюють у 20 балів тільки за умови надання вичерпної відповіді на запитання.
11–15	Досить повна, без суттєвих неточностей, наведена у логічно правильній послідовності відповідь, яка свідчить про ґрунтовні та систематизовані знання матеріалу навчальної дисципліни; демонструє здатність студента впевнено оперувати здобутими знаннями: відтворювати та аналізувати отриману інформацію, пояснювати основні закономірності, робити висновки, чітко висловлювати думку та формулювати відповідь. Як правило, таку оцінку отримує студент, який відповів на запитання на 70–90 %.
6–10	Не зовсім повна, із неточностями та окремими незначними помилками, наведена в основному у правильній послідовності відповідь, яка свідчить про задовільні знання матеріалу навчальної дисципліни, демонструє здатність студента відтворювати основний матеріал навчальної дисципліни відповідно до поставленого запитання. Як правило, таку оцінку отримує студент, який відповів на запитання на 50–70 %.
1–5	Фрагментарна, із суттєвими неточностями та принциповими помилками відповідь, яка свідчить про неповноту знань основного матеріалу навчальної дисципліни, демонструє наявність у студента утруднень при відтворенні інформації відповідно до поставленого запитання.

	Як правило, таку оцінку отримує студент, який відповів на запитання менше ніж на 50 %.
0	Відповідь не надано або надана відповідь не відповідає поставленому запитанню.

Оцінка атестаційного екзамену від 0 до 59 балів вважається незадовільною.

Завідувач кафедри
інформаційних систем і технологій
д-р фіз.-мат. наук, професор



Валерій ГАВРИЛЕНКО

ДОДАТОК А
ФОРМА БІЛЕТА АТЕСТАЦІЙНОГО ЕКЗАМЕНУ
 НАЦІОНАЛЬНИЙ ТРАНСПОРТНИЙ УНІВЕРСИТЕТ
АТЕСТАЦІЙНИЙ ЕКЗАМЕН

*Освітньо-професійна програма «Інженерія програмного забезпечення»
 спеціальності 121 «Інженерія програмного забезпечення»
 Перший (бакалаврський) рівень вищої освіти*

ЗАТВЕРДЖУЮ

Проректор
з навчальної роботи

Завідувач кафедри
інформаційних систем і технологій

Білет № ____

Запитання І рівня складності

Запитання та варіанти відповідей	Позначення студентом вибраної відповіді
1. Текст запитання	
а) варіант відповіді	
б) варіант відповіді	
в) варіант відповіді	
2. Текст запитання	
а) варіант відповіді	
б) варіант відповіді	
в) варіант відповіді	
3. Текст запитання	
а) варіант відповіді	
б) варіант відповіді	
в) варіант відповіді	
4. Текст запитання	
а) варіант відповіді	
б) варіант відповіді	
в) варіант відповіді	
5. Текст запитання	
а) варіант відповіді	
б) варіант відповіді	
в) варіант відповіді	

6. Текст запитання	
а) варіант відповіді	
б) варіант відповіді	
в) варіант відповіді	
7. Текст запитання	
а) варіант відповіді	
б) варіант відповіді	
в) варіант відповіді	
8. Текст запитання	
а) варіант відповіді	
б) варіант відповіді	
в) варіант відповіді	
9. Текст запитання	
а) варіант відповіді	
б) варіант відповіді	
в) варіант відповіді	
10. Текст запитання	
а) варіант відповіді	
б) варіант відповіді	
в) варіант відповіді	

Запитання II рівня складності

11. Текст запитання

12. Текст запитання

13. Текст запитання

14. Текст запитання

Розглянуто та схвалено на засіданні кафедри інформаційних систем і технологій.

Протокол № 8 від 29 березня 2022 року.

Розглянуто та схвалено на засіданні Вченої Ради факультету транспортних та інформаційних технологій.

Протокол № 8 від 26 квітня 2022 року.

Розглянуто та схвалено на засіданні Науково-методичної ради Національного транспортного університету.

Протокол № 29 від 29 квітня 2022 року.