

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ



ЗАТВЕРДЖУЮ

Проректор
роботи

з науково-педагогічної

Віктор ЛОПАТОВСЬКИЙ

2025р.

ПРОГРАМА ФАХОВОГО ІСПИТУ

для вступу на навчання для здобуття ступеня вищої освіти «магістр» на
основі раніше здобутого ступеня вищої освіти бакалавра, магістра (освітньо-
кваліфікаційного рівня спеціаліста)

Галузь знань: F Інформаційні технології

Спеціальність: F2 Інженерія програмного забезпечення

Освітня програма: Інженерія програмного забезпечення

Схвалено на засіданні кафедри

протокол № 8 від 13 04 2025 р.

Зав. кафедри

Леонід БЕДРАТЮК

Гарант ОП

Леонід БЕДРАТЮК

Програма розглянута та схвалена на засіданні вченої ради факультету
інформаційних технологій протокол № 9 від 17.04 2025 р.

Голова вченої ради ФІТ

Тятяна ГОВОРУЩЕНКО

2. Загальні положення

Вступ на навчання на освітню програму "Інженерія програмного забезпечення" за спеціальністю F2 "Інженерія програмного забезпечення" для здобуття ступеня вищої освіти "магістр" проводиться на базі раніше здобутого ступеня вищої освіти бакалавра, магістра або освітньо-кваліфікаційного рівня спеціаліста.

Перелік освітніх компонентів (навчальних дисциплін), на базі яких складається іспит:

- Конструювання програмного забезпечення,
- Моделювання та оцінка програмного забезпечення,
- Архітектура та проектування програмного забезпечення,
- Аналіз вимог до програмного забезпечення,
- Базис даних,
- Об'єктно-орієнтоване програмування.

Мета вступного фахового іспиту полягає у перевірці здатності до опанування освітньої програми "Інженерія програмного забезпечення" магістерського рівня вищої освіти на основі здобутих раніше компетентностей.

3. Технологія проведення фахового іспиту.

Іспит проводиться у формі тестування із комп'ютерною обробкою результатів. Система проведення вступних іспитів є оригінальною розробкою ХНУ і захищена свідоцтвом про авторське право № 39534 від 08.08.2011 р. Вона розроблена на підставі таких документів: Закону України «Про вищу освіту», «Положення про приймальну комісію ХНУ», Порядку прийому до вищих навчальних закладів України та Правил прийому до Хмельницького національного університету.

Основні положення системи тестування із комп'ютерною обробкою результатів викладені нижче. Бази даних тестових завдань створюються для всіх дисциплін, з яких проводиться тестування, щорічно поповнюються і вдосконалюються.

Бази даних тестових завдань або навчальні програми, за якими вони створені, є відкритими. Університет щорічно оприлюднює їх у паперовому або в електронному вигляді.

Відповідальність за зміст і якість тестових завдань покладається на завідувача випускової кафедри та гаранта освітньої програми.

Для автоматизованого формування білетів використовують комплекс комп'ютерних програм, які компонують бази даних тестових завдань з кожної дисципліни, формують екзаменаційні білети за допомогою випадкової вибірки та роздруковують їх.

Екзаменаційні білети, що включають тестові завдання, формують і тиражують комп'ютерними засобами перед початком тестування. Білет включає 50 тестових завдань. Сформовані білети засвідчуються печаткою приймальної комісії.

Номер кожного екзаменаційного білета збігається з номером талона відповідей, який додається до нього.

Організація автоматизованого формування комплекту екзаменаційних білетів до вступних іспитів, контроль за ним покладається на відповідального секретаря Приймальної комісії або його заступника.

Тестування проводиться відповідно до розкладу в аудиторіях, що обладнані необхідними технічними засобами.

Пропуск вступників до аудиторії тестування проводить відповідальний секретар ПК та його заступники. При цьому перевіряється паспорт та перепустка, у якій вказана особа вступника, дата і час тестування.

Кожний учасник тестування витягує номер, який вказує його місце в аудиторії. Всі місця за столами пронумеровані.

В аудиторії тестування дозволяється присутність громадських спостерігачів (у т.ч. з числа батьків вступників).

Вступникам видаються титульні листи і проводиться роз'яснення щодо їх заповнення.

Після розміщення учасників тестування в аудиторії вступники особисто вибирають екзаменаційні білети, що розкладені на столі.

Після отримання екзаменаційних білетів вступники працюють над розв'язанням тестових завдань протягом встановленого часу.

Талони відповідей надаються кожному вступнику в одному екземплярі. Забороняється видача вступнику другого талона. Талон відповідей заповнюється вступником відповідно до роз'яснення щодо їх заповнення.

Після закінчення роботи над тестами, або добігання до кінця часу, відведеного на тестування, вступники здають підписані роботи разом з талонами відповідей, які до початку сканування знаходяться на столі екзаменатора.

Сканування талонів відповідей починається після здачі робіт всіма вступниками у їх присутності. Процес сканування талонів відповідей демонструється за допомогою проектору на великому екрані.

Після закінчення сканування та комп'ютерної обробки талонів відповідей результати тестування демонструються на екрані у вигляді екзаменаційної відомості, в якій відсутні прізвища вступників, а є лише номер екзаменаційного білета. Далі персонал приймальної комісії вносить в комп'ютер інформацію про відповідність номера екзаменаційного білета прізвищу вступника. На екрані демонструється екзаменаційна відомість з прізвищами вступників, яка роздруковується і завіряється відповідальним секретарем приймальної комісії.

Критерії оцінювання та фахового іспиту особливості переведення кількості правильних відповідей у 200-бальну шкалу затверджуються на засіданні Приймальної комісії та наводяться в додатку до Правил прийому.

Перелік освітніх компонентів (навчальних дисциплін), на базі яких складається іспит

Конструювання програмного забезпечення

Вступ в конструювання програмного забезпечення. Історія розвитку. Базові терміни. Елементи конструювання ПЗ. Ключові принципи конструювання. Основи конструювання програмного забезпечення. Планування та управління процесом розроблення та супроводу програмного забезпечення. Технологія конструювання програмного забезпечення. Базові терміни. Види програмного забезпечення. Життєвий цикл процесу розробки ПЗ. Макетування. Стратегії конструювання ПЗ. Високоякісне кодування. Уніфікована мова моделювання UML. Структура мови UML. Предмети, відношення в UML. Діаграми, їх класифікація в UML. Загальні правила і механізми мови моделювання. Моделювання класів. Діаграми класів. Об'єкти і відносини класів. Реалізація програмою на псевдокоді. Діаграми класів. Керування конструюванням. Моделювання структури. Об'єктно-орієнтовне моделювання поведінки. Кінцеві автомати. Поведінка додатків. Засоби моделювання поведінки. Моделювання взаємодії та поведінки системи. Діаграми станів. Діаграми діяльності. Діаграми взаємодії, кооперації. Вплив UML на процес розробки. Підвищення продуктивності програмування. Рівні

модельовання. Основні концепції і фази проектування. Методологія і технологія розробки ПЗ. Методологія RAD - Rapid Application Development. Проектування в конструюванні. Кодування. Загальні принципи використання змінних. Проектування в конструюванні. Методика оптимізації коду. Проектування в конструюванні. Патерни (шаблони проектування). Патерн Фасад. Патерн Спостерігач. Якість програмного забезпечення.

Література:

1. Gerardus Blokdyk. UML: A Complete Guide – 2020 Edition. 5STARCook, 2020. – 297 с
Постіл. С.Д. UML. Уніфікована мова модельовання інформаційних систем: Навч. посіб., 2019. - 321 с.
2. Piotr Sliż. Organizacja procesowo-projektowa. Istota, modelowanie, pomiar dojrzałości. Difin – 2021 – 292 str.
3. Ian Sommerville. Software Engineering, 10th edition. Published by Pearson . July 14th 2021 – 816 p.
4. Philippe Kruchten. The Rational Unified Process: An Introduction (5rd Edition) (Addison-wesley Object Technology Series). Addison-Wesley Professional; 5rd edition. 2018 – 407 p.
5. Pinto J. Project Management: Achieving Competitive Advantage, 5th Edition. – 2020. – 592 p.
6. Martin Fowler. Refactoring: Improving the Design of Existing Code (Web Edition), 3rd Edition/ Web Edition.- 2022

Модельовання та оцінка програмного забезпечення

Теоретичні аспекти процесу модельовання ПЗ. Поняття моделі, процесу модельовання ПЗ. Етапи процесу модельовання. Принципи структурних методів та структурного аналізу. UML – складова процесу розробки ПЗ.. Концептуальна модель UML. Види діаграм UML. Правила UML. Основи структурного модельовання ПЗ. Модельовання класів, об'єктів та екземплярів: класи, атрибути, операції, обов'язки; модельовання словника системи; модельовання розподілу обов'язків у системі; модельовання непрограмних сутностей; створення якісних абстракцій. Основні механізми доповнення й розширення будівельних блоків UML. Модельовання зв'язків і відношень.

Типові зв'язки: зв'язки: залежності, узагальнення, асоціації; модельовання простих залежностей; модельовання одиничного спадкування; модельовання структурних зв'язків; створення мереж зв'язків. Розширені зв'язки. Інтерфейси, типи та ролі. Діаграми для модельовання статичних характеристик системи. Моделі статичних представлень системи. Діаграми класів. Діаграми об'єктів. Компоненти та інтерфейси. Діаграми компонентів. Модельовання поведінки ПЗ. Модельовання варіантів використання: дійові особи, включення й розширення. Модельовання поведінки елемента. Реалізація варіантів використання за допомогою кооперацій. Діаграми

варіантів використання. Моделювання контексту системи. Моделювання вимог до системи. Пряме й зворотне проектування. Моделювання взаємодій об'єктів. Стратегія моделювання взаємодій як обміну повідомленнями між об'єктами. Ролі, посилення, повідомлення, дії та послідовності. Моделювання потоків керування. Створення добре структурованих алгоритмів. Діаграми взаємодії. Моделювання потоків керування за часом. Моделювання потоків керування по організації. Діаграми діяльності. Моделювання потоку робіт. Моделювання операції.

Моделювання подій сигналів. Події сигналу, виклику, часу і зміни. Моделювання серії сигналів. Опрацювання подій в активних і пасивних об'єктах. Моделювання винятків. Кінцеві автомати. Моделювання поведінки працюючих спільно об'єктів. Стани, переходи і діяльності. Моделювання ЖЦ об'єкта. Створення добре структурованих алгоритмів. Процеси і потоки керування. Активні об'єкти, процеси і потоки. Моделювання множини потоків керування. Моделювання міжпроцесорної комунікації. Побудова абстракцій захищених потоків. Моделювання систем реального часу. Час, тривалість і місце розташування. Моделювання тимчасових обмежень. Моделювання розподілу об'єктів. Діаграми станів. Моделювання реактивних об'єктів. Пряме і зворотне проектування. Основи моделювання архітектури ПЗ. Поняття про архітектуру та раціональний уніфікований процес. Моделювання архітектурних зразків. Зразки і каркаси. Моделювання зразків проектування. Моделювання архітектурних зразків. Забезпечення доступності зразків. Моделювання кооперації. Кооперації, реалізації й взаємодії. Моделювання реалізації варіанта використання. Моделювання реалізації операції. Моделювання механізму. Моделювання пакетів як способів організації елементів моделі. Пакети, видимість, імпорт і експорт. Моделювання груп елементів. Моделювання архітектурних виглядів. Масштабування великих систем. Паттерни проектування. Поняття про патерни (шаблони проектування). Види патернів. Патерн Фасад. Патерн Спостерігач.

Література:

1. Stephen H. Kan. *Metrics and Models in Software Quality Engineering*, 2nd Edition. Addison-Wesley Professional, 2020. – 560
2. Paul Clements, Rick Kazman, Mark Klein. *Evaluating Software Architectures: Methods and Case Studies*. Addison-Wesley Professional, 2020. – 368 с.
3. Edmund M. Clarke, Orna Grumberg, Daniel Kroening, Doron Peled, Helmut Veith. *Model Checking*, 2nd Edition. MIT Press, 2020. – 424 с.
4. Fabio Paternò. *Model-Based Design and Evaluation of Interactive Applications*. Springer, 2020. – 320 с
5. A. Wayne Wymore. *Model-Based Systems Engineering*. CRC Press, 2020. – 732 с

6. Li, S., Yuan, H., Rusconi, P. (2020). Cognitive Modeling for Automated Human Performance Evaluation at Scale. (n.p.): Springer International Publishing

Архітектура та проектування програмного забезпечення

Поняття архітектури програми. Цілі вибору архітектури. Декомпозиція. Моделі, каркаси та зразки проектування. Використання моделей.. Каркаси. Класифікація архітектур. Зразки проектування. Компоненти. Типи архітектур та їх моделі. Архітектури, засновані на потоках даних. Незалежні компоненти. Віртуальні машини. Репозиторні архітектури. Рівневі архітектури. Додатки зі змішаною архітектурою. Процедура вибору архітектури. Архітектура: нотація, стандарти та інструментальні засоби. Нотація. Інструментальні засоби. Стандарт IEEE/ANSI для опису проектів. Контроль якості при виборі архітектури. Якість та вибір архітектури. Вибір з альтернативних архітектур. Перевірка архітектури з допомогою діаграм варіантів використання. Інспектування вибору архітектури. Вплив вибору архітектури на SPMR. Вступ в детальне проектування. Поняття детального проектування. Співвідношення варіантів використання, архітектури та детального проектування. Типова схема процесу детального проектування. Проектування по схемі USDP. Проектування та інтерфейси. Повторно використовуємі елементи. Діаграми послідовності та діаграми потоків даних в детальному проектуванні. Детальні діаграми послідовності. Детальні діаграми потоків даних. Специфікація алгоритмів, класів та функцій. Інваріанти класу. Інваріанти, передумови та післяумови функцій. Блок-схеми. Псевдокод. Використання блок-схем та псевдокоду. Зразки проектування: прийоми детального проектування. Креаційні зразки проектування. Структурні зразки проектування. Зразки проектування, засновані на поведінці додатка. Бібліотеки стандартних шаблонів. Стандарти, нотація та інструментальні засоби проектування. Бібліотека стандартних шаблонів (STL) C++. Стандарт IEEE 890. Інструменти, що використовують вихідний код: Javadoc. Вплив детального проектування на проект. Оцінка об'єму робіт з допомогою детального проектування. Якість і метрики в детальному проектуванні. Інспектування для детального проектування.

Література:

1. Len Bass, Paul C. Clements, Rick Kazman. Software Architecture in Practice, 4th Edition. Addison-Wesley Professional, 2021. – 464 с.

2. Humberto Cervantes, Rick Kazman. Designing Software Architectures: A Practical Approach, 2nd Edition. Addison-Wesley Professional, 2024. – 352 с.
3. Фрімен Е. Head First. Патерни проєктування /Ерік Фрімен, Елізабет Робсон. – Харків: Фабула, 2020. – 672 с.
4. Richards M. Fundamentals of Software Architecture: An Engineering Approach / Mark Richards, NealFord. – Sebastopol, California: O'Reilly Media – 1st edition, 2020. – 419 p.
5. Lanciaux R. Modern Front-end Architecture: Optimize Your Front-end Development with Components, Storybook, and Mise en Place Philosophy / Ryan Lanciaux – New York: Apress, 2021 – 144 p.
6. Frighi V. Smart Architecture – A Sustainable Approach for Transparent Building ComponentsDesign / Valentina Frighi – Berlin: Springer – 1st edition, 2022. – 293 p.

Аналіз вимог до програмного забезпечення

Поняття інформаційної системи. Означення ІС. Класифікація ІС: по масштабу, архітектурі, характеру використання інформації, системі представлення даних, по стандартам управління і технологіям комунікації які підтримуються, степені автоматизації. Роль вимог в задачі впровадження АІС. Методи класифікації вимог. Означення поняття вимог. Класифікація вимог: вимога до продукту і процесу, рівні вимог, системні вимоги і вимоги до програмного забезпечення, функціональні, нефункціональні вимоги і характеристики продукту, класифікація RUP. Методології і стандарти які регламентують роботу з вимогами. Властивості вимог. Повнота, ясність коректність і узгодженість(несуперечливість), верифікація (придатність до перевірки), необхідність і корисність при експлуатації, здійснюваність (виконуваність), трасування, впорядкованість по важливості і стабільності, наявність кількісної метрики, вимоги яких не повинно бути. Процес аналізу вимог. Робочий потік аналізу вимог, для чого потрібно аналізувати вимоги, хто створює і використовує вимоги, організація роботи з вимогами на прикладі MSF Контекст задачі аналізу вимог. Аналіз вимог, бізнес-аналіз , аналіз проблемної області. Роль глосарія при АТ. Методології бізнес-аналізу. Вимоги і архітектура АІС. Аналіз вимог і інші робочі потоки програмної інженерії Виявлення вимог. Джерело вимог. Стратегії виявлення вимог. Що потрібно пам'ятати при опитуванні. Формування бачення. Введення продукту. Границі проекту. Бачення RUP. Бачення/рамки MSF. Класифікація і специфікація вимог. Актори і варіанти використання. Глосарій. Специфікація варіанту використання. Специфікація не функціональних вимог. Атрибути вимог. Розширений аналіз вимог. Моделювання. Вибір моделей. Моделі UML, які пояснюють функціональність системи. Діаграми UML, які пояснюють внутрішню будову системи. Альтернативні мови моделювання. Розширений

аналіз вимог. Ілюстровані сценарії і прототипи. Цілі прототипування. Класифікація прототипування. Ілюстровані сценарії прецедентів. Середня інтенсивність використання. Документування вимог. Документування вимог в RUP. Документування вимог на основі міжнародного стандарту ISO / IEC / IEEE. Вимоги до зовнішнього інтерфейсу. Інші нефункціональні вимоги. Документування вимог в MSF. Перевірка вимог. Верифікація і валідація. Деякі типові проблемні ситуації процесу формування оцінки вимог. Методи і засоби перевірки вимог. Управління вимогами. Принципи і прийоми керування вимогами. Управління змінами. Трасування вимог. Удосконалення процесів роботи з вимогами. Моделі вдосконалення. Принципи вдосконалення. Процес вдосконалення. Оцінка результатів і прийняття рішень. Вимоги до управління проектом. Планування проекту на основі RUP. Вимоги в гнучких методологіях. Аналіз вимог і керування ризиками. Стратегії і роботи по управлінню ризиком. Аналіз сучасного розвитку АІС. Критерії вибору ПЗ. Процес вибору.

Література:

1. Phillip A. Laplante, Neeraj Kumar, A. Kassab. *Requirements Engineering for Software and Systems*, 4th Edition. CRC Press, 2023. – 400 с.
2. Nazim H. Madhavji, Liliana Pasquale, Alessio Ferrari, Stefania Gnesi (Eds.). *Requirements Engineering: Foundation for Software Quality – REFSQ 2020*. Springer, 2020. – 267 с
3. Ralph R. Young. *The Requirements Engineering Handbook*. Artech House, 2020. – 312 с
4. Gerardus Blokdyk. *Requirement Engineering: A Complete Guide – 2020 Edition*. 5STARCook, 2020. – 304 с.
5. Fred Heath, *Managing Software Requirements the Agile Way: Bridge the gap between software requirements and executable specifications to deliver successful projects*, Packt Publishing Ltd, 2020 - 214 p.

Бази даних

Предмет, об'єкт, метод, мета і задачі дисципліни "Бази даних". Базові визначення і поняття. Призначення й основні компоненти системи баз даних. Існуючі вимоги до організації баз даних. Основна література по дисципліні і її короткий аналіз. Бази даних. Функції СКБД. Файлові системи. Структури файлів. Іменування файлів. Захист файлів. Режим багатокористувацького доступу. Області застосування файлів. Споживачі інформаційних систем. Основні функції СКБД. Типова організація сучасної СКБД. Ієрархічні системи. Ієрархічні структури даних. Маніпулювання даними. Обмеження цілісності. Мережеві системи. Мережні структури даних. Маніпулювання даними.

Обмеження цілісності. Переваги і недоліки ранніх підходів до реалізації
Загальні поняття реляційного підходу до організації БД. Основні концепції і
терміни. Базові поняття реляційних баз даних. Тип даних. Домен. Схема
відношень, схема бази даних, кортеж, відношення. Фундаментальні
властивості відношень. Реляційна модель даних Загальна характеристика.
Цілісність сутності і посилань Базові поняття реляційної моделі даних.
Множини Операції над множинами. Декартовий добуток множин.
Відношення. Приклади відношень. Бінарні відношення (відношення степеня 2).
Відношення еквівалентності. Відношення порядку. Функціональне
відношення, n-арне відношення (відношення степеня n). Транзитивне
замикання відношень. Загальна характеристика реляційної моделі даних. Типи
даних. Прості типи даних. Структуровані типи даних. Посилальні типи даних.
Типи даних, використовувані в реляційній моделі. Домени. Відношення,
атрибути, кортежі відношення. Визначення і приклади. Властивості
відношень. Перша нормальна форма Реляційна алгебра. Огляд реляційної
алгебри. Замкнутість реляційної алгебри Відношення, сумісні по типі.
Оператор перейменування трибутів. Теоретико-множинні оператори.
Об'єднання. Перетинання. Вирахування. Декартовий добуток. Спеціальні
реляційні оператори. Вибірка (обмеження, селекція). Проекція. З'єднання.
Загальна операція з'єднання. Тета-з'єднання. Екві-з'єднання. Природне
з'єднання. Розподіл. Приклади використання реляційних операторів. Залежні
реляційні оператори. Оператор з'єднання. Оператор перетинання. Оператор
розподілу. Примітивні реляційні оператори. Оператор декартового добутку.
Оператор проекції. Оператор вибірки Оператори об'єднання і вирахування.
Запити, невимовні засобами реляційної алгебри. Погана нормалізація
відношень. Невимовність транзитивного замикання реляційними
операторами. Крос-таблиці. Мови запитів до баз даних. Мова SQL. Оператори
SQL. Оператори DDL (Data Definition Language) оператори визначення
об'єктів бази даних. Оператори DML (Data Manipulation Language) оператори
маніпулювання даними. Оператори захисту і керування даними. Приклади
використання операторів маніпулювання даними. INSERT вставка рядків у
таблицю. UPDATE - відновлення рядків у таблиці. DELETE видалення рядків
у таблиці. Приклади використання оператора SELECT. Вибір даних з однієї
таблиці. Вибір даних з декількох таблиць. Використання імен кореляції
(альясів, псевдонімів). Використання агрегатних функцій у запитах.
Використання агрегатних функцій з угрупованнями. Використання підзапитів.
Використання об'єднання, перетинання і різниці. Синтаксис оператора вибірки
даних (SELECT)BNF-нотація. Синтаксис оператора вибірки. Синтаксис
з'єднаних таблиць Синтаксис умовних виражень роздїгнула WHERE. Порядок
виконання оператора SELECT. Реалізація реляційної алгебри засобами
оператора SELECT (Реляційна повнота SQL. Оператор декартового добутку.
Оператор проекції. Оператор вибірки. Оператор об'єднання. Оператор
вирахування. Оператор з'єднання. Оператор перетинання. Оператор
розподілу.

СКБД MySQL Загальна характеристика СКБД MySQL. Об'єкти БД: таблиці, форми, запити, звіти. Структура СКБД MySQL. Створення БД. Створення таблиць БД. Типи та властивості полів БД. Заповнення, редагування таблиць. Ключові поля. Відношення між таблицями. Створення схеми даних. СКБД Access (Open Office.org Base). Загальна характеристика СКБД MsAccess (Open Office.org Base).

Нормальні форми відношень. Етапи розробки бази даних. Критерії оцінки якості логічної моделі даних. Адекватність бази даних предметній області. Легкість розробки і супроводження бази даних. Швидкість операцій відновлення даних (вставка, відновлення, видалення). Швидкість операцій вибірки даних. 1НФ (Перша Нормальна Форма). Аномалії оновлення. Аномалії вставки (INSERT). Аномалії оновлення (UPDATE). Аномалії видалення (DELETE). Функціональні залежності. Визначення функціональної залежності. Функціональні залежності відношень і математичне поняття функціональної залежності. 2НФ (Друга Нормальна Форма). Аналіз декомпозованих відношень (аномалії вставки, оновлення та аномалії видалення, що залишилися для 2НФ), 3НФ (Третя Нормальна Форма). Алгоритм нормалізації (приведення до 3НФ). Аналіз критеріїв для нормалізованих і ненормалізованих моделей даних. Порівняння нормалізованих і ненормалізованих моделей. OLTP і OLAP-системи. Коректність процедури нормалізації декомпозиція без утрат. Теорема Хеза. Нормальні форми більш високих порядків. НФБК (Нормальна Форма Бойса-Кодда). 4НФ (Четверта Нормальна Форма). 5НФ (П'ята Нормальна Форма). Алгоритм нормалізації для 5НФ. Моделі даних. Основні поняття ER-діаграм. Приклад розробки простої ER-моделі. Концептуальні і фізичні ER-моделі. Транзакції і цілісність баз даних. Приклад порушення цілісності бази. Поняття транзакції. Обмеження цілісності. Класифікація обмежень цілісності. Класифікація обмежень цілісності за способами реалізації. Класифікація обмежень цілісності за часом перевірки. Класифікація обмежень цілісності по області дії. Обмеження домену. Обмеження атрибута. Обмеження кортежу. Обмеження відношення. Обмеження бази даних. Реалізація декларативних обмежень цілісності засобами SQL. Загальні принципи реалізації обмежень засобами SQL. Синтаксис обмежень стандарту SQL. Синтаксис операторів SQL, що використовують обмеження. Транзакції і паралелізм. Робота транзакцій у суміші. Проблеми паралельної роботи транзакцій. Проблема втрати результатів відновлення. Проблема незафіксованої залежності (читання «брудних» даних, неакurate зчитування). Проблема несумісного аналізу. Неповторювальне зчитування. Фіктивні елементи (фантоми). Несумісний аналіз. Конфлікти між транзакціями. Блокування. Рішення проблем паралелізму за допомогою блокувань. Види відновлення даних. Індивідуальний відкат транзакції. Відновлення після м'якого збою. Відновлення після твердого збою.

СКБД в архітектурі "клієнт-сервер". Архітектура "клієнт-сервер". Відкриті системи. Клієнти і сервери локальних мереж. Системна архітектура "клієнт-сервер". Сервери баз даних. Взаємодія різних типів СКБД. Технології

доступу та обміну даних між різними типами СКБД. Розробка мережевих баз даних. Проблеми розробки та способи проектування «клієнт-серверних» додатків база даних. Розподілені бази даних. Різновиди розподілених систем БД. Іменування об'єктів і організація розподіленого каталогу. Розподілена компіляція запитів. Керування транзакціями і синхронізація. Інтегровані чи федеративні системи і мультибази даних. Системи керування базами даних наступного покоління. Орієнтація на розширену реляційну модель. Абстрактні типи даних. Генерація систем баз даних, орієнтованих на додатки. Оптимізація запитів, керована правилами. Підтримка історичної інформації і темпоральних запитів. Часові бази даних. Проблеми часу в базах даних. Зв'язок часових баз даних СКБД із загальними поняттями об'єктно-орієнтованого підходу та реляційними моделями. Приклади мов програмування ООБД. Мови запитів часових баз даних.

Література:

1. Abraham Silberschatz, Henry F. Korth, S. Sudarshan. *Database System Concepts*, 7th Edition. McGraw-Hill Education, 2020. – 1376
2. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom. *Database Systems: The Complete Book*, 2nd Edition. Pearson, 2020. – 1248 с.
3. Alan Beaulieu *Learning SQL: Generate, Manipulate, and Retrieve Data*/Beaulieu Alan - O'Reilly Media 2020. - 377 pages.
4. Carlos Coronel. *Database Systems: Design, Implementation, & Management* (MindTap Course List) / Coronel Carlos, Morris Steven – Boston, Massachusetts: Cengage Learning – 14th edition, 2022 – 816 p.
5. Ying Bai. *SQL Server Database Programming with Visual Basic.NET*/Ying Bai – Wiley, 2020– 688p.
6. Nenad Jukić, Susan Vrbsky, Svetlozar Nestorov, Abhishek Sharma. *Database Systems: Introduction to Databases and Data Warehouses*, 2nd Edition. Prospect Press, 2020. – 512 с

Об'єктно-орієнтоване програмування

Технології об'єктно-орієнтованого проектування програмних систем. Парадигми програмування. Об'єктно-орієнтований підхід до програмування. Фундаментальні поняття ООП. Критика ООП. Сучасні технології та платформи проектування програмних систем. Особливості створення додатків в .NetFrameWork та особливості мови С#. Базові поняття .NetFrameWork. Створення додатків з допомогою .NetFramework. Класи та об'єкти С#. Основні поняття. Оголошення та структура класу. Дані та методи класу. Декларації private, protected, public. Ініціалізація об'єктів класу. Конструктори та деструктори, їх призначення, оголошення, та варіанти викликів. Звичайні,

констатні та статичні дані, методи, особливості їхнього оголошення. Одержання детальної інформації про клас.

Абстрактні класи. Розгорнуті й вказівкові типи. Класи й структури. Структури. Вбудовані структури. Перерахування. Особливості перерахувань. Концепція наслідування. Відношення між класами. Відношення «є» і «має». Успадкування. Механізм успадкування. Позитивні аспекти успадкування. Позитивні аспекти успадкування і їх практичні можливості. Реалізація успадкування в C#. Додавання полів нащадком. Конструктори батьків і нащадків. Додавання методів і зміна методів батька. Статичний контроль типів і динамічне зв'язування. Інтерфейси. Означення Інтерфейсу. Два способи реалізації інтерфейсу. Перетворення до класу інтерфейсу. Вбудовані інтерфейси.

Множинне наслідування. Проблеми множинного спадкування. Колізія імен. Успадкування від загального батьківського класу. Делегати. Означення делегата. Функції вищих порядків. Операції над делегатами. Клас Delegate. Основні методи й властивості класу Delegate. Приклад використання делегатів. Події. Події для багатоадресної передачі. Порівняння методів екземплярів зі статичними методами, що використовуються в якості обробника подій. Використання подійних засобів доступу. Рекомендації по обробці подій в середовищі .NET Framework. Використання вбудованого делегата EventHandler. Використання подій. Приклади використання. Універсальність. Класи з родовими параметрами. Спадкування і універсальність. Синтаксис універсального класу. Клас з універсальними методами. Два основних механізми об'єктної технології. Стек. Від абстрактного, універсального класу до конкретних версіями. Обмежена універсальність. Синтаксис обмежень. Список з можливістю пошуку елементів за ключем. Родове породження класу. Універсальність і спеціальні випадки класів. Універсальні структури. Універсальні інтерфейси. Універсальні делегати. .Net Framework і універсальність. Коректність і стійкість програмних систем. Життєвий цикл програмної системи. Три закони програмотехніки. Перший закон програмотехніки (закон для розробника). Другий закон програмотехніки (закон для користувача). Третій закон програмотехніки (закон того, хто вичікує). Налагодження. Створення надійного коду. Мистецтво налагодження. Налагодження і умовна компіляція. Класи Debug. Метод Флойда та затвердження тверджень. Класи StackTrace і BooleanSwitch. Налагодження та інструментальне середовище Visual Studio. Обробка виняткових ситуацій. Обробка виключень у мовах C/C++. Схема обробки виключень в C#. Усунення винятків. Створення об'єктів винятків. Захоплення винятку. Паралельна робота обробників виключень. Блокування. Схема Бертрана обробки виняткових ситуацій. Клас винятків.

Література:

1. Andrew Troelsen, Philip Japikse. *Pro C# 10 with .NET 6: Foundational Principles and Practices in Programming*, 11th Edition. Apress, 2022. – 1 632

2. David Flanagan JavaScript: The Definitive Guide: Master the World's Most-Used Programming Language/Flanagan David - O'Reilly Media 2020. - 704 p.
3. Joel Murach ASP.NET Core MVC/Murach Joel, Dalamater Mary - Mike Murach & Associates 2020. - 780 p.
4. Dan Clark. *Beginning C# Object-Oriented Programming: Learn Object-Oriented Programming with C#*, 2nd Edition. Apress, 2021. – 452 c.
5. Raihan Taher. *Hands-On Object-Oriented Programming with C#: Write maintainable, scalable, and testable software using C# 10 and .NET 6*. Packt Publishing, 2022. – 452 c
6. Mark J. Price. *C# 9 and .NET 5 – Modern Cross-Platform Development: Build applications with C#, .NET Core, Entity Framework Core, ASP.NET Core, and Blazor*. Packt Publishing, 2020. – 752 c.
7. Herbert Schildt Java: The Complete Reference, Twelfth Edition/Schildt Herbert - McGraw Hill, 2021. - 1280 p.

5. Переведення тестових балів

Кожен екзаменаційний білет включає 50 тестових завдань, кожне із яких оцінюється одним балом. Таблиця переведення правильних відповідей у 200-бальну шкалу, наведена нижче.

Таблиця переведення тестових балів (правильних відповідей) тесту з іноземної мови та фахового іспиту при вступі на ОС «Магістр», що проводяться в Хмельницькому національному університеті до шкали 100–200

Кількість тестових балів (правильних відповідей)*	Бал за шкалою 100–200
1-4	Не складено
5	100
6	104
7	108
8	112
9	115
10	118
11	121
12	124
13	127
14	130
15	132
16	134
17	136
18	138

Кількість тестових балів (правильних відповідей)*	Бал за шкалою 100–200
28	151
29	152
30	153
31	154
32	155
33	156
34	157
35	158
36	159
37	161
38	163
39	165
40	167
41	169
42	171

19	140
20	142
21	144
22	145
23	146
24	147
25	148
26	149
27	150

43	174
44	177
45	180
46	184
47	188
48	192
49	196
50	200

* усі білети містять 50 питань з однією правильною відповіддю.