

Голова Приемальної комісії
Ректор



Факультет інформатики та обчислювальної техніки

повна назва факультету/навчально-наукового інституту

ПРОГРАМА фахового іспиту

для вступу на освітньо-професійні, освітньо-наукові програми підготовки магістра

за спеціальністю 121 Інженерія програмного забезпечення

Інженерія програмного забезпечення інформаційних систем

Інженерія програмного забезпечення комп'ютерних систем

123 Комп'ютерна інженерія

Комп'ютерні системи та мережі

126 Інформаційні системи та технології

Інтегровані інформаційні системи, Інформаційні управляючі системи та технології,

Інформаційне забезпечення роботехнічних систем,

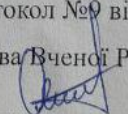
Інформаційні системи та технології

Програму ухвалено:

Вченою Радою Факультету інформатики та
обчислювальної техніки

Протокол №9 від 25 березня 2024 р.

Голова Вченої Ради

 д.т.н., проф. С. СТИПЕНКО

Київ – 2024

ВСТУП

Програма фахового іспиту передбачає перевірку набуття вступником компетентностей та результатів навчання, що визначені стандартом вищої освіти за спеціальностями 121 Інженерія програмного забезпечення, 123 Комп'ютерна інженерія, 126 Інформаційні системи та технології для першого (бакалаврського) рівня вищої освіти за освітньо-професійними та освітньо-науковими програмами підготовки магістра «Інженерія програмного забезпечення інформаційних систем», «Інженерія програмного забезпечення комп'ютерних систем», «Комп'ютерні системи та мережі», «Інтегровані інформаційні системи», «Інформаційні управляючі системи та технології», «Інформаційне забезпечення роботехнічних систем», «Інформаційні системи та технології».

На навчання для здобуття ступеня магістра приймаються особи, які здобули ступінь бакалавра або освітній ступінь магістра/спеціаліста.

Проведення фахового іспиту має такі цілі:

– забезпечити рівні можливості випускникам вищих навчальних закладів для вступу на навчання за освітньо-професійною та освітньо-науковою програмами підготовки магістрів;

– перевірити рівень теоретичних знань та професійних навичок абітурієнтів, вміння використовувати їх при вирішенні конкретних професійних завдань.

Фаховий іспит проводиться у формі письмового екзамену.

До екзаменаційного білету фахового іспиту включаються як теоретичні так і практичні завдання. До складу програми увійшли теми з Проекту Програми предметного тесту з інформаційних технологій єдиного фахового вступного випробування для вступу на навчання для здобуття ступеня магістр на основі НРК6, НРК7.

Кожний екзаменаційний білет фахового іспиту містить п'ять завдань.

Тривалість фахового іспиту – 180 хв.

ПЕРЕЛІК НАВЧАЛЬНОГО МАТЕРІАЛУ, ЯКИЙ ВІНОСИТЬСЯ НА ФАХОВИЙ ІСПИТ

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. Кортежі, 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
- 2. Архітектура обчислювальних систем
 - 2.1 Функції бінарної логіки
 - 2.2 Представлення 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 та доповнювальному кодах.
 - 2.2.2. Принципи зображення дійсних чисел в пам'яті програми у форматі з плаваючою комою. Основні арифметичні операції над дійсними числами у форматі з плаваючою комою та їх 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 Переферійні пристрої
- 3. Бази даних 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 та 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)
- 4. Інженерія систем і програмного забезпечення
 - 4.1. Складні та великі системи
 - 4.1.1. Властивості систем: емерджентність, адитивність, еквіфінальність
 - 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

Об'єктно-орієнтоване проектування

Архітектурне проектування

Інтерфейсне проектування

4.5.2. Парадигми проектування: функціональна декомпозиція згори вниз, 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. Налагодження: Точки зупинки (Breakpoints), Спостереження за змінними (Variable Watch), Виведення на консоль (Console Output), Налагоджувач (Debugger), Аналізатори коду (Code Analyzers)
- 4.6.4. Керування конфігурацією програмного забезпечення та контроль версій
- 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 (водопадна), ітераційна, інкрементна
 - 4.8.2. Промислові технології розробки. 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.

- 5. Кібербезпека та захист інформації
 - 5.1. Основи кібербезпеки
 - 5.1.1. Поняття кіберпростору та інформаційного простору
 - 5.1.2. Інформаційна безпека як сфера національної безпеки України
 - 5.1.3. Поняття кібербезпеки, захисту інформації та кіберзахисту
 - 5.1.4. Види захисту інформації
 - 5.1.5. Поняття конфіденційності, цілісності, доступності
 - 5.1.6. Принципи кібербезпеки
 - 5.2. Кіберзагрози та кібератаки
 - 5.2.1. Поняття загроз, атак, вразливості
 - 5.2.2. Класифікація загроз, атак
 - 5.2.3. Кіберзлочини. Кібервійна. Кібероборона
 - 5.2.4. Кібертероризм. Кіберрозвідка
 - 5.2.5. Модель порушника
 - 5.2.6. Поняття, сутність та основні завдання комплексної системи захисту інформації
 - 5.3. Безпека мережі
 - 5.3.1. Поняття про шкідливе програмне забезпечення
 - 5.3.2. Шпигунські програми, фішинг, програми-вимагачі.
 - 5.3.3. DDoS-атаки.
- 6. Математика в ІТ
 - 6.1. Застосування методів математичного аналізу, аналітичної геометрії, лінійної алгебри в ІТ
 - 6.1.1. Числова послідовність та її границя. Нескінченно малі та великі величини. Порівняння нескінченно малих і великих величин

- 6.1.2. Похідна та її застосування для дослідження функцій однієї змінної
- 6.1.4. Невизначені, визначені інтеграли: поняття та застосування
- 6.1.5. Функції багатьох змінних. Частинні похідні. Необхідні і достатні умови екстремуму, умовного екстремуму
- 6.1.6. Метод найменших квадратів (лінійна залежність)
- 6.1.7. Числові ряди. Поняття їх збіжності
- 6.1.8. Основні означення теорії диференціальних рівнянь: порядок диференціального рівняння, частинний розв'язок, загальний розв'язок, загальний інтеграл, задача Коші
- 6.2. Елементи аналітичної геометрії
 - 6.2.1. Пряма і площина в просторі. Поняття гіперплощини
 - 6.2.2. Криві другого порядку. Еліпс, гіпербола, парабола та їх властивості
 - 6.2.3. Поняття поверхні, її типи
- 6.3. Елементи лінійної алгебри
 - 6.3.1. Матриці. Дії з матрицями. Визначники. Обернена матриця
 - 6.3.2. Власні вектори та власні числа матриці
 - 6.3.3. Системи лінійних алгебраїчних рівнянь, умови їх розв'язуваності. Методи їх розв'язання
 - 6.3.4. Лінійний векторний простір та його основні властивості. Розмірність і базис простору
- 6.4. Методи оптимізації
 - 6.4.1. Основні поняття та цілі в задачах лінійного та нелінійного програмування. Градієнтний метод: ідея та алгоритм
- 6.5. Дискретна математика
 - 6.5.1. Множини. Поняття чітких та нечітких множин. Операції над чіткими множинами: об'єднання, перетин, різниця, доповнення,

булеан множини, декартів добуток

- 6.5.2. Бінарні відношення та їх властивості: рефлексивність, симетричність, транзитивність
- 6.5.3. Комбінаторний аналіз. Правило суми та добутку. Сполуки, перестановки, розміщення: без повторень та з повтореннями. Принцип включень і виключень
- 6.5.4. Елементи математичної логіки. Пропозиційна логіка. Логіка висловлювань. Логічні сполучники. Атомарні формули. Таблиці істинності
- 6.5.5. Графи. Типи графів: Орієнтовні та неорієнтовні граfi. Вершини та ребра, степінь вершини, суміжність. Ізоморфізм графів. Операції над графами: об'єднання, пряма сума, доповнення, вилучення ребра, вилучення вершини
- 6.5.6. Маршрути, ланцюги, цикли та їх різновиди у графах
- 6.5.7. Зв'язність графів, компоненти зв'язності неорієнтованих графів. Відстань між вершинами
- 6.5.8. Древа, ліси: основні поняття
- 6.6. Основний понятійний апарат теорії ймовірностей та математичної статистики
 - 6.6.1. Стохастичний експеримент. Простір елементарних подій. Операції над подіями. Класична, геометрична, статистична, аксіоматична ймовірність. Умовні ймовірності
 - 6.6.2. Формула повної ймовірності. Формула Байєса. Схема незалежних випробувань Бернуллі. Закон великих чисел
 - 6.6.3. Одновимірні дискретні випадкові величини. Числові характеристики дискретних випадкових величин. Моменти дискретних випадкових величин

- 7 Мережі та обмін даними
- 7.1. Класифікація та функції комп'ютерних мереж. Комутація каналів та комутація пакетів. Топології комп'ютерних мереж
- 7.2 Поняття протоколу та інтерфейсу, ієрархія протоколів, потік інформації в мережі. Еталонні моделі ISO/OSI та TCP/IP
- 7.3. Інтернет речей: основні поняття, сфери застосування.
- 8. Операційні системи
- 8.1. Призначення операційних систем
- 8.1.1. Різноманітність операційних систем (однокористувацькі, багатокористувацькі, реального часу)
- 8.1.2. Основні функції операційних систем
- 8.1.3. Вимоги до операційних систем, поняття відмовостійкості
- 8.2. Файлові системи
- 8.2.1. Основні поняття про файли і файлові системи
- 8.2.2. Логічна та фізична організація файлів
- 9. Основи програмування
- 9.1 Сутність і види мов програмування
- 9.1.1. Поняття класу та об'єкта в об'єктно-орієнтованому програмуванні; конструктор і деструктор, інтерфейс та реалізація
- 9.1.2. Базові концепції об'єктно-орієнтованого програмування: абстракція, інкапсуляція, спадкування, поліморфізм
- 9.1.3. Зв'язки між класами в об'єктно-орієнтованому програмуванні: асоціація, агрегація, композиція, спадкування, залежність, реалізація
- 9.1.4. Порівняння процедурного та об'єктно-орієнтованого програмування.

- 9.2. Принципи та сфера застосування видів програмування: функціональне, логічне, подійне-орієнтоване, реактивне, узагальнене програмування
- 9.3. Моделі паралельних обчислень: класифікація Флінна
- 9.3.2. Ефективність та вартість паралельних обчислень
- 9.4. Трансляція та виконання: компілятор, інтерпретатор, компоувальник.
- 10. Штучний інтелект
- 10.1. Фундаментальні поняття
- 10.2. Пошук у просторі станів
- 10.2.1. Стратегії пошуку: пошук в ширину, пошук в глибину, двонаправлений пошук.
- 10.3. Машинне навчання
- 10.3.1. Задача класифікації. Навчання з вчителем та без учителя.
- 10.3.2. ВИБур тренувальних та валідаційних даних для навчання
- 10.3.3. Поняття: штучний нейрон, штучна нейронна мережа, функції активації штучного нейрона (лінійна, порогова, сигмоїдна, радіально-базисна Гауса).

ПРИКЛАД ЕКЗАМЕНАЦІЙНОГО БІЛЕТУ

Завдання 1. (max 20 балів)

Поняття, сутність та основні завдання комплексної системи захисту інформації.
(В описі обов'язкове наведення прикладів).

Завдання 2. (має 20 балів)

Класифікація та функції комп'ютерних мереж. Комутація каналів та комутація пакетів. Топології комп'ютерних мереж. (В описі обов'язкове наведення прикладів).

Завдання 3. (має 20 балів)

Ідентифікація класів предметної області. UML-діаграми ієрархії класів: моделювання підсистем, класів та зв'язків між ними. (В описі обов'язкове наведення прикладів).

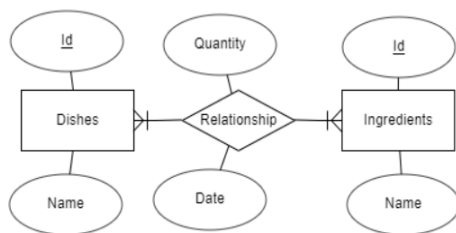
Завдання 4. (має 20 балів)

На мові програмування C, C++, C# або Java написати фрагмент програми (функцію, метод або клас), для розв'язання наступної задачі:

Задано одновимірні масиви $A(n)$ та $B(n)$. Сформувані одновимірний масив $C(2*n)$, елементами якого є елементи масивів $A(n)$ та $B(n)$, записані через один ($a_1, b_1, a_2, b_2, a_3, b_3, \dots$).

Завдання 5. (має 20 балів)

1)



2)

3)

Для наведеної ER діаграми створіть наступні скрипти та запит:

a) DDL (CREATE, ALTER) – створення таблиць, формалізація зв'язку, внесення необхідних обмежень цілісності

b) DML (INSERT) – додайте по 7 записів до кожної таблиці. UPDATE – оновіть записи в асоціативній таблиці, змініть дату на DateNew в тих записах, де назви інгредієнтів починаються з символів XXX (довільно за Вашим вибором).

с) SELECT – оберіть всі інгредієнти, які були додані до блюд ДО певної дати Date1 (довільно за Вашим вибором).

ПРИКІНЦЕВІ ПОЛОЖЕННЯ

При проведенні фахового іспиту заборонено користуватись будь-якими допоміжними матеріалами, гаджетами тощо.

Фаховий іспит здатності вступника:

- узагальнювати отримані знання для вирішення конкретних практичних завдань;
- застосовувати правила, методи, принципи, закони у конкретних ситуаціях;
- інтерпретувати схеми, графіки, діаграми;
- застосовувати підходи до моделювання та проектування програмного забезпечення у відповідно до вказаної предметної області;
- втілювати архітектурні шаблони для вказаної предметної області;
- викладати матеріал логічно, послідовно.

Фаховий іспит складається з п'яти завдань.

Критерії оцінювання завдань фахового іспиту враховують наступне:

- оцінка за виконання фахового іспиту виставляється за системою ECTS - 100-бальна шкала;
- максимальна кількість балів, яка нараховується за виконання окремого завдання 1 – 20 балів, завдання 2 – 20 балів, завдання 3 – 20 балів, завдання 4 – 20 балів, завдання 5 – 20 балів.
- оцінювання результатів кожного завдання здійснюється у п'ятирівневій системі балів (таблиця 1).

Таблиця 1. Критерії оцінювання завдань

Оцінка	Опис
16-20	Наведені необхідні теоретичні відомості (основні теоретичні

	положення методу розв'язання задачі), повністю виконано практичне завдання
10-14	Наведені необхідні теоретичні відомості (основні теоретичні положення методу розв'язання задачі), є деякі несуттєві недоліки при виконанні практичного завдання
6-9	Наведені деякі теоретичні відомості (деякі теоретичні положення методу розв'язання задачі), повністю виконано практичне завдання
1-5	Наведені деякі теоретичні відомості (деякі теоретичні положення методу розв'язання задачі) без виконання практичного завдання
0	Теоретичні відомості відсутні, практичне завдання не виконано

Загальний критерій оцінюється по сумі балів за відповіді на завдання із всіх п'яти розділів, максимальна оцінка складає 100 балів:

$$R_{\text{заг}} = R_{\text{розд1}} + R_{\text{розд2}} + R_{\text{розд3}} + R_{\text{розд4}} + R_{\text{розд5}} =$$

$$20 + 20 + 20 + 20 + 20 = 100 \text{ балів}$$

«Правила прийому до КПІ ім. Ігоря Сікорського в 2024 році» вимагають при обчисленні конкурсного балу застосування шкали оцінювання 100...200 балів, перерахунок в яку зі стобальної шкали РСО відбувається відповідно до таблиці.

Таблиця відповідності оцінок РСО (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

СПИСОК ЛІТЕРАТУРИ

1. Базилевич Л. Є. Дискретна математика у прикладах і задачах : підручник / Л. Є. Базилевич — Львів : Видавець І. Е. Чижигов, 2013. — 487 с.
2. Бандура В. В. Архітектура та проектування програмного забезпечення: конспект лекцій / В. В. Бандура, Р. І. Храбатин; ІФНТУНГ. — Івано-Франківськ: ІФНТУНГ, 2012. — 240 с.
3. Васильєв О. Програмування С++ в прикладах і задачах : навч. посіб. / О. Васильєв. — Київ : Видавництво Ліра-К, 2017. — 382 с.
4. Ковалюк Т. В. Основи програмування / Т. В. Ковалюк. — Київ : Видавнича група ВНУ, 2005. — 384 с.
5. Кулаков Ю. О. Комп'ютерні мережі: підручник / Ю. О. Кулаков, Г. М. Луцький; — Київ : «Юніор», 2005. — 396 с.
6. Кулаков Ю. О. Комп'ютерні мережі : навч. посіб. / Ю. О. Кулаков, І. А. Жуков. — Київ : Вид-во Нац. Авіа. Ун-ту «НАУ-друк», 2009. — 329с.
7. Лавріщева К. М. Програмна інженерія / К. М. Лавріщева. — Київ : Видавнича група ВНУ, 2007. — 413 с.
8. Левус Є. В. Вступ до інженерії програмного забезпечення : навч. посіб. / Є. В. Левус, Н. Б. Мельник. — Львів : Видавництво Львівської політехніки, 2018. — 246 с.
9. Матвієнко М. П. Дискретна математика : підручник / М. П. Матвієнко — 2-ге вид., перероб. і доп. — Київ : Ліра-К, 2017. — 324 с.
10. Основи дискретної математики / Ю. В. Капітонова [та ін.]. — Київ : Наукова думка, 2002. — 560 с.
11. Нікольський Ю. В. Дискретна математика / Ю. В. Нікольський, В. В. Пасічник, Ю. М. Щербина — Київ : Видавнича група ВНУ, 2007. — 368 с.
12. Порєв В. М. Об'єктно-орієнтоване програмування. Конспект лекцій [Електронний ресурс] : навчальний посібник для здобувачів ступеня бакалавра за освітньою програмою «Інженерія програмного забезпечення комп'ютерних систем» спеціальності 121 «Інженерія програмного забезпечення» / Порєв В. М. ; КПІ ім. Ігоря Сікорського. — Електронні текстові дані (1 файл: 4,8 МБайт). — Київ : КПІ ім. Ігоря Сікорського, 2022. — 271 с. — Режим доступу: <https://ela.kpi.ua/handle/123456789/51571>.
13. Andrew S. Tanenbaum. Computer Networks, 6th edition. Andrew S. Tanenbaum, Nick Feamster, David J. Wetherall. : GLOBAL EDITION, 2021. — 946 P.
14. Bjarne Stroustrup. Programming: Principles and Practice Using C++ (2nd Edition) [Online] Available from: — Режим доступу: <https://dl.icdst.org/pdfs/files3/fef0590f02fa06bb42cba558fbc9e51c.pdf>
15. C# documentation. [Online] Available from: — Режим доступу: <https://learn.microsoft.com/en-us/dotnet/csharp/>

16. C++ Standard Library including C++ 14 & C++ 17 [Online] Available from: – Режим доступу: <https://www.educative.io/courses/cpp-standard-library-including-cpp-14-and-cpp-17>
17. Gerbert Schildt. Java: A Beginner’s Guide / Gerbert Schildt : McGrawHil (Eighth Edition), 2018. – 814 P.
18. Martin Fowler. UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd Edition / Martin Fowler : Addison-Wesley, Inc. 2003. – 208 P.
19. UML 2.5.1 Офіційна нотація [Електронний ресурс]. – Режим доступу: <https://www.omg.org/spec/UML/About-UML/>
20. Using Rational RequisitePro. [Електронний ресурс]. – Режим доступу: <http://pic.dhe.ibm.com/infocenter/rqmhelp/v2r0>.
21. Jacobson I. Object-Oriented Software Engineering. A use Case Driven Approach, Revised Printing / I. Jacobson. – New York: Addison-Wesley Publ. Co., 1994. – 529 P.
22. Lynn Bailey. Learn SQL / Lynn Bailey : Publisher: Peter Inc. 2012. — 592 P.
23. Hector Garcia-Molina. Database Systems. The Complete Book. Second Edition / H. Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom: Pearson Education Inc. 2009. — 1240 с.

РОЗРОБНИКИ ПРОГРАМИ

д.т.н. професор
кафедри ОТ

Михайло НОВОТАРСЬКИЙ

к.т.н. доцент
кафедри ОТ

Віктор ПОРЄВ

к.т.н. доцент
кафедри ОТ

Олександр РОКОВИЙ

к.т.н. доцент
кафедри ІСТ

Максим БУКАСОВ

к.т.н. доцент
кафедри ІСТ

Ксенія УЛЬЯНИЦЬКА

к.т.н. доцент
кафедри ІІІ

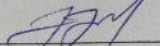
Катерина ЛІЩУК

ст. викладач
кафедри ІІІ

Олена МАРЧЕНКО

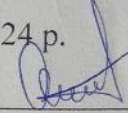
Програму рекомендовано кафедрою інформатики та програмної інженерії

Протокол №12 від 14.03.2024 р.

завідувач кафедри _____  Едуард ЖАРИКОВ


Програму рекомендовано кафедрою обчислювальної техніки

Протокол № 9 від 06.03.2024 р.

завідувач кафедри _____  Сергій СТИПЕНКО

Програму рекомендовано кафедрою інформаційних систем та технологій

Протокол №10 від 13.03.2024 р.

завідувач кафедри _____  Олександр РОЛІК