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

Затверджую

Голова Приймальної комісії
Ректор

Михайло ЗГУРОВСЬКИЙ

підпис *дата* 27.04.2024

Факультет біомедичної інженерії

повна назва факультету/навчально-наукового інституту

ПРОГРАМА
фахового іспиту
для вступу на освітньо-професійну програму підготовки магістра
«Комп'ютерні технології в біології та медицині»
за спеціальністю 122 Комп'ютерні науки

Програму ухвалено:
Вченою Радою факультету біомедичної інженерії
Протокол № 8 від 25 березня 2024 р.
Голова Вченої Ради
_____ Віталій МАКСИМЕНКО

Київ – 2024

ВСТУП

Програма фахового іспиту призначена для оцінювання якості підготовки вступників при вступі на навчання на освітньо-професійну програму підготовки магістра «Комп'ютерні технології в біології та медицині» за спеціальністю 122 Комп'ютерні науки.

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

Проведення фахового іспиту має на меті:

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

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

Фаховий іспит спрямований на перевірку здатностей 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 відповідно до вказаної предметної області;

– втілювати архітектурні шаблони для вказаної предметної області;

– викладати матеріал логічно, послідовно.

Фаховий іспит складається з п'яти завдань та проводиться у формі письмового екзамену. До екзаменаційного білету фахового іспиту включаються як теоретичні, так і практичні завдання. До складу програми увійшли теми з Проєкту програми предметного тесту з інформаційних технологій єдиного фахового вступного випробування для вступу на навчання для здобуття ступеня

магістр на основі НРК6, НРК7.

Тривалість фахового іспиту – 180 хв. Диференціації робочого часу, відведеного на виконання кожного завдання, немає. Фіксується час початку і закінчення роботи.

ПЕРЕЛІК НАВЧАЛЬНОГО МАТЕРІАЛУ, ЯКИЙ ВИНОСИТЬСЯ 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 складності

1.1.2. Поняття абстрактного типу даних.

Стандартні абстрактні типи даних: стеки, списки, вектори, словники, множини, мультимножини, черги, черги з пріоритетами

1.1.3. Кортежі, множини, словники, одно- та двозв'язні списки.

Реалізація абстрактних типів даних з оцінюванням складності операцій

1.1.4. Базові алгоритми та їх складність: пошук, сортування (прості сортування видаленням, вставками, обмінами та удосконалені сортування деревом, сортування Шела, 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 «зменшуй та володарюй» та приклади застосування. Стратегія «розділяй та володарюй» та приклади застосування. Стратегія балансування та приклади застосування.

Динамічне програмування та приклади застосування.

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, напівроз'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. Системи числення, двійкове, вісімкове, шістнадцядкове числення. Представлення чисел у цілочисельному форматі та форматі із плаваючою комою. Доповнювальний код

2.2.2. Основні арифметичні операції над цілочисельними двійковими числами. Формат чисел з плаваючою комою, переваги та недоліки, основні арифметичні операції та їх проблеми

2.3. Пристрої введення-виведення. Поняття шини комп'ютера

2.4. Функціональна організація

2.4.1. Структура комп'ютера, класична архітектура фон Неймана. Види пам'яті: кеш-пам'ять, оперативна пам'ять, зовнішня пам'ять. CPU, GPU. Периферійні пристрої

3. Бази даних та сховища даних

3.1. Ключі та нормалізація даних: основні нормальні форми (1NF, 2NF, 3NF, BCNF, 4NF)

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)

3.7. Розподілені бази даних/хмарні обчислення: доступність, масштабованість, виклики, технології

3.8. 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).

4. Інженерія систем і програмного забезпечення

4.1. Складні системи

4.1.1. Класифікація систем за призначенням, взаємодією із зовнішнім середовищем, походженням, видом елементів, способом орґанізації

4.1.2. Складні та великі системи. Властивості та характерні особливості складних систем

4.1.3. Поняття системи та її структури

4.1.4. Поняття декомпозиції та агрегування

4.2. Моделі систем

4.2.1. Моделювання систем

4.2.2. Зв’язок між системою та моделлю. Ізо- та гомоморфізм

4.2.3. Класифікація моделей систем

4.3. Аналіз вимог

4.3.1. Джерела та методи збирання вимог

4.3.2. Вимоги користувача: модель вимог на основі прецедентів (варіантів використання) (Use Case Diagram), історії користувачів (user story). Вимоги до описів варіантів використання

4.3.3. Класифікація вимог до програмного забезпечення: функціональні та нефункціональні вимоги, обмеження, структуризація функціональних вимог.

4.4. Проєктування програмного забезпечення

4.4.1. Моделювання проєкту з UML: діаграми статичні та динамічні, логічні та фізичні

4.4.2. Види проєктування: архітектурне (верхній рівень) та деталізоване проєктування (класів, атрибутів, операцій), проєктування інтерфейсу користувача

4.4.3. Парадигми проєктування: функціональна декомпозиція згори вниз, архітектура, орієнтована на дані, об'єктно-орієнтований аналіз та проєктування, подієво-керована архітектура

4.4.4. Ідентифікація класів предметної області. UML-діаграми ієрархії класів: моделювання підсистем, класів та зв’язків між ними

4.4.5. Проєктування сценаріїв реалізації варіантів використання на основі UML-діаграм послідовностей та комунікації

4.4.6. Роль архітектури. Стандартні архітектури: клієнт-серверна та n-рівнева архітектура, Model View Controller

4.4.7. Архітектурні моделі та патерни проєктування (Abstract Factory, Facade, Decorator, Flyweight, Visitor, Observer, Proxy,Strategy, Chain of Responsibility)

4.5. Реалізація програмного забезпечення

4.5.1. Вимоги до оформлення коду: стиль, розбиття на структуровані одиниці, найменування змінних, класів, об’єктів тощо

4.5.2. Засоби автоматичної ґенерації програмного коду на основі трансформацій UML-модель – код ООП-мовою, повторне використання коду ПЗ

4.5.3. Налагодження: Точки зупинки (Breakpoints), Спостереження за змінними (Variable Watch), Виведення на консоль (Console Output), Налагоджувач (Debugger), Аналізатори коду (Code Analyzers)

4.5.4. Керування конфігурацією програмного забезпечення та контроль версій

4.5.5. Постійна інтеграція/постійне впровадження (Continuous Integration/Continuous Delivery)

4.6. Забезпечення якості: тестування, верифікація, валідація

4.6.1. Призначення, спільне та відмінності процесів тестування, верифікації, валідації

4.6.2. Види тестів: модульні, інтеграційні, регресійні, системні, валідаційні

4.6.3. Тестування методами білої та чорної скрині

4.6.4. Розробка через тестування (Test-driven development)

4.6.5. Додаткові техніки верифікації та валідації: інспекція коду, перевірка на відповідність стандартам і вимогам, оцінювання зручності використання та користувацького досвіду, перевірка продуктивності та масштабованості

4.7. Командна робота, підходи до розробки програмного забезпечення (ПЗ)

4.7.1. Класичні моделі розробки ПЗ: каскадно-водопадна, ітераційна, інкрементна

4.7.2. Промислові технології розробки. RUP- уніфікований підхід, керований варіантами використання, архітектурно-центрований, ітераційний, інкрементний

4.7.3. Ролі та обов'язки у програмній команді, переваги командної роботи, ризики та складність такої співпраці

4.7.4. Технології гнучкої розробки ПЗ та їх особливості: Agile, Scrum, Extreme Programming (XP), Kanban

4.7.5. Моделі керування командною роботою (на основі UML діаграм Ґанта і Перта)

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

5.3.1.1. Поняття про шпигунські програми, фішинг, програми-вимагачі, DDoS-атаки, соціальну інженерію

5.3.2. Способи забезпечення безпеки мережі

5.3.2.1. Поняття про брандмауери, контроль доступу, сегментацію мережі

5.3.2.2. Системи виявлення та запобігання вторгненням

5.3.3. Процедури ідентифікації, автентифікації, авторизації користувачів мережі

5.3.3.1. Поняття процедур ідентифікації, автентифікації та авторизації користувача

5.3.3.2. Види ідентифікації, автентифікації та авторизації користувача

5.4. Криптографічні методи захисту даних

5.4.1. Класифікація методів криптографічного захисту за призначенням

5.4.2. Сутність криптографії, криптології, стеганографії

5.4.3. Поняття моделі симетричної криптосистеми

5.4.4. Шифри перестановки (шифр перестановки за ключем), шифр простої заміни (шифр Цезаря), шифр складної заміни (шифр Віженера)

5.4.5. Поняття симетричних блочних алгоритмів шифрування

5.4.6. Поняття моделі асиметричної криптосистеми

5.4.7. Поняття процесу автентифікації документів. Електронний підпис

6. Математика в IT

6.1. Основний понятійний апарат математичного аналізу

6.1.1. Числова послідовність та її границя. Нескінченно малі та великі величини.

Порівняння нескінченно малих і великих величин

6.1.2. Функції однієї змінної. Границя функції в точці. Неперервні функції

6.1.3. 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 графи. 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. Дерева, 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

6.6.4. Закони розподілу неперервних випадкових величин: рівномірний, нормальний

6.6.5. Багатовимірні дискретні величини та їх числові характеристики.

Коефіцієнт кореляції

6.6.6. Поняття випадкової функції і процесу

6.6.7. Основні задачі математичної статистики. Первинна обробка даних. Числові характеристики вибіркової сукупності

6.6.8. Статистичний та інтервальний ряди розподілу. Гістограма

6.6.9. Точкові та інтервальні оцінки. Довірчі інтервали

6.6.10. Перевірка статистичних гіпотез. Критерії згоди. Критерій Пірсона

7. Мережі та обмін даними

7.1. Основні поняття та функції комп'ютерних мереж

7.1.1. Класифікація та функції комп'ютерних мереж. Комутація каналів та комутація пакетів. Топології комп'ютерних мереж

7.1.2. Поняття протоколу та інтерфейсу, ієрархія протоколів, потік інформації в мережі. Еталонні моделі ISO/OSI та TCP/IP

7.2. Типи мережевих сервісів, сервіси зі встановленням з'єднань та без встановлення з'єднань. Основні функції. Поняття портів та сокетів. Протоколи TCP та UDP.

7.3. Маршрутизація та обмін даними

7.3.1. Функції мережевого рівня, задача маршрутизації. Протокол IP. IP-адреси та їх властивості

7.4. Технології бездротових мереж

7.4.1. Принципи організації бездротових мереж

7.4.2. Технології Wi-Fi

7.5. Безпека мережі: на канальному рівні, VLAN. VPN. Основні захищені мережеві протоколи

8. Операційні системи

8.1. Призначення операційних систем

8.1.1. Різноманітність операційних систем (однокористувацькі, багатокористувацькі, реального часу, вбудовані системи)

8.1.2. Основні функції операційних систем

8.1.3. Вимоги до операційних систем, поняття відмовостійкості

8.2. Принципи побудови операційної системи

8.2.1. Типи архітектур ядра операційної системи

8.2.2. Привілейований режим і режим користувача

8.2.3. Системні виклики

8.3. Одночасність

8.3.1. Мультизадачність

8.3.2. Мультипроцесорність

8.3.3. Паралельність

8.4. Модель процесу

8.4.1. Блок керування процесом

8.4.2. Контекст процесу

8.4.3. Стани процесу

8.4.4. Розподіл пам'яті (типи адрес, методи розподілу пам'яті)

8.4.5. Віртуальна пам'ять (сторінкова, сегментна, сегментно-сторінкова організація пам'яті, свопінг)

8.5. Файлові системи

8.5.1. Основні поняття про файли і файлові системи

8.5.2. Логічна та фізична організація файлів

8.6. Поняття системи реального часу

8.6.1. Визначення систем реального часу, основні характеристики

8.6.2. Види систем реального часу та їх відмінності

9. Основи програмування

9.1 Об'єктно-орієнтоване програмування

9.1.1. Поняття класу та об'єкта; конструктор і деструктор, інтерфейс та реалізація

9.1.2. Базові концепції ООП: абстракція, інкапсуляція, спадкування, поліморфізм

9.1.3. Зв'язки між класами: асоціація, агрегація, композиція, спадкування, залежність, реалізація

9.2. Принципи та сфера застосування видів програмування: функціональне,

логічне, подійне, реактивне, генеративне програмування

9.3. Паралельні та розподілені обчислення

9.3.1. Моделі паралельних обчислень

9.3.2. Ефективність та вартість паралельних обчислень

9.3.3. Закон Амдаля

9.3.4. Синхронне та асинхронне програмування

9.4. Трансляція та виконання

9.4.1. Компілятор, інтерпретатор, компонувальник, компілятор в байт-код або проміжний код, JIT компілятор, система виконання (Runtime)

9.4.2. Форма Backus–Naur (БНФ) та розширена нотація БНФ

9.4.3. Регулярні вирази

10. Штучний інтелект

10.1. Фундаментальні поняття

10.1.1. Інтелект, штучний інтелект, поняття агента і середовища, задачі штучного інтелекту, раціональність, сильний і слабкий штучний інтелект, ризики штучного інтелекту

10.2. Пошук у просторі станів

10.2.1. Стратегії пошуку: пошук в ширину, пошук в глибину, двонаправлений пошук, жадібний алгоритм.

10.3. Основи подання знань

10.3.1 Факти, знання, властивості знань. Моделі знань: семантичні мережі, фрейми, логічні моделі, продукційні правила

10.4. Машинне навчання

10.4.1. Навчання з вчителем та без, навчання з підкріпленням, регресійні і класифікаційні задачі

10.4.2. Лінійна і логістична регресія: ідентифікація, регуляризація, сфера застосування

10.4.3. Поняття: формальний нейрон, штучна нейронна мережа, функції активації формального нейрона

ПРИКЛАД ЕКЗАМЕНАЦІЙНОГО БІЛЕТУ

Завдання 1.

П'ять пакетних задач A, B, C, D, E надходять у систему практично одночасно. Очікується, що час їхнього виконання складе 3, 5, 2, 7 і 1 хв відповідно. Їхні встановлені пріоритети складають 3, 4, 2, 5 і 1, причому 5 – найвищий пріоритет. Визначте середній оборотний час для алгоритмів планування: 1) з урахуванням пріоритетів, 2) "першим прийшов – першим обслугований", 3) "найкоротша задача – перша", зневажаючи часом, що витрачається на переключення між процесами. Передбачається, що в кожен момент часу запущена одна задача, що працює аж до завершення. Усі задачі обмежені тільки можливостями процесора.

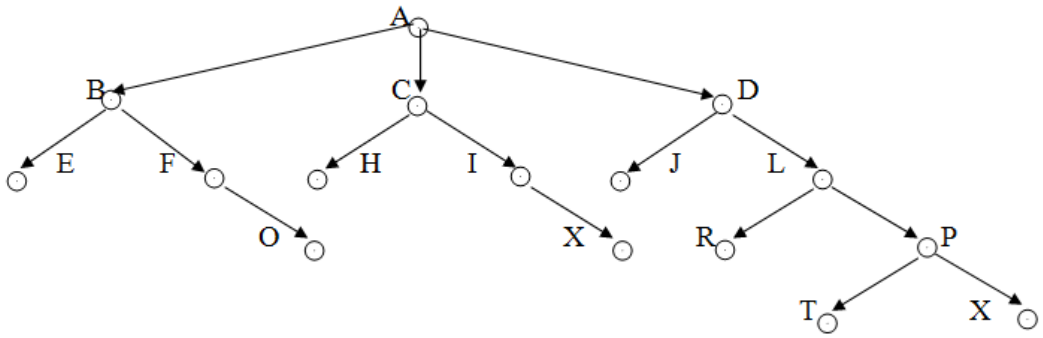
Завдання 2.

Визначити для наступного графу:

- послідовність вершин, які будуть перевірятися при визначення маршруту між вершинами A і X за стратегією «Підіймання на пагорб» (Hill climbing);

- маршрут, який першим буде винайдено за цим методом.

Оцінювальна функція вершини x: *h(x) = 1/N,* де N – номер в латинському алфавіті букви, якою позначена вершина графа.



Завдання 3.

Побудувати інтерполяційний многочлен Лагранжа для функції $f(x)=e^{-x}$, якщо вузлами інтерполяції є точки $x_1=1$, $x_2=2$, $x_3=3$. Оцінити похибку для $x=1{,}5$.

Завдання 4.

Нижче наведено код, в якому описуються класи для реалізації умовної “програми-дієтолога” (автоматичний підрахунок калорій залежно від кількості вжитих продуктів та способів їх приготування). В коді бракує кількох методів. Ієрархія наслідування класів страв (англ. dish) включає “просту страву” (BasicDish) з одного продукту та “складну (комплексну) страву” (ComplexDish), реалізовану через патерн “компонувальник” (англ. “composite”), до складу якої входять інші страви. До інтерфейсу страви входять операції зі зміни ваги та питомої калорійності на 100 г. продукту, а також операцій отримання ваги та загальної калорійності страви. Реалізація обробки страв виконана на основі патерну “відвідувач” (англ. “visitor”), для чого створено ієрархію “кухонного приладдя” (англ. kitchen tool). Реалізовано два “кухонні прилади”: “сковорідка“ для смаження страв (нехай вона збільшує калорійність страви в півтора рази та зменшує вагу на 20%) та “аналізатор вмісту” для виведення на екран звіту з калорійності страви та її складових. **Завдання**:

1) Намалювати **UML діаграму**, що включає усі класи в коді, за виключенням стандартних (таких, як std::string).

2) Написати **код двох пропущених методів** для зчитування та зміни ваги для класу ComplexDish (вага має змінюватися пропорційно для всіх складових, тобто якщо вага усієї страви збільшилась вдвічі, то це означає, що вага кожної складової має збільшитись удвічі). Інший код за межами даних методів дописувати не можна.

3) Написати, що буде **виведено на екран** в процесі виконання програми.

```
#include <iostream>
#include <string>
#include <list>

// forward declarations of dish types
class BasicDish;
class ComplexDish;

// interfaces
class IKitchenTool {
    public:
```

```
        virtual void process(BasicDish* dish) = 0;
        virtual void process(ComplexDish* dish) = 0;
};

class IDish {
    public:
        virtual std::string getDescription() = 0;
        virtual float getWeight() = 0;
        virtual float getCalories() = 0;
        virtual void setWeight(float w) = 0;
        virtual void setCaloriesPerWeight(float cpw) = 0;
        virtual void apply(IKitchenTool* tool) = 0;
};
// classes
class AbstractNamedDish: public IDish {
    public:
        AbstractNamedDish(std::string desc):
            description(desc) {
        }
        virtual std::string getDescription() override {
            return description;
        }
    private:
        std::string description;
};

class BasicDish: public AbstractNamedDish {
    public:
        BasicDish(std::string desc, float w, float cpw):
            AbstractNamedDish(desc), weight(w), caloriesPerWeight(cpw) {
        }
        virtual float getWeight() override {
            return weight;
        }
        virtual float getCalories() override {
            return caloriesPerWeight * weight / 100;
        }
        virtual void setWeight(float w) override {
            weight = w;
        }
        virtual void setCaloriesPerWeight(float cpw) override {
            caloriesPerWeight = cpw;
        }
        virtual void apply(IKitchenTool* tool) override {
            tool->process(this);
        }
    private:
        float weight;
        float caloriesPerWeight;
};

class ComplexDish: public AbstractNamedDish {
    public:
        ComplexDish(std::string desc):
            AbstractNamedDish(desc) {
        }
        void addDish(IDish* sub_dish) {
            dishes.push_back(sub_dish);
```

```
        }
        std::list<IDish*>& dishList() {
            return dishes;
        }
        virtual float getCalories() override {
            float cal = 0;
            for (auto sub_dish: dishes) {
                cal += sub_dish->getCalories();
            }
            return cal;
        }
        virtual void setCaloriesPerWeight(float cpw) override {
            float cpwKoef = cpw / (getCalories() / getWeight());
            for (auto sub_dish: dishes) {
                sub_dish->setCaloriesPerWeight(sub_dish->getCalories() /
sub_dish->getWeight() * cpwKoef);
            }
        }
        virtual void apply(IKitchenTool* tool) override {
            tool->process(this);
        }
    private:
        std::list<IDish*> dishes;
};

class FryingPan: public IKitchenTool {
    public:
        virtual void process(BasicDish* dish) override {
            dish->setCaloriesPerWeight(1.5 * dish->getCalories() / (dish-
>getWeight() / 100));
            dish->setWeight(0.8 * dish->getWeight());
        }
        virtual void process(ComplexDish* dish) override {
            process((BasicDish*) dish);
        }
};

class DishAnalyzer: public IKitchenTool {
    public:
        virtual void process(BasicDish* dish) override {
            std::cout << dish->getDescription()
                << " has total " << dish->getCalories() << " kcal"
                << " in " << dish->getWeight() << " g" << std::endl;
        }
        virtual void process(ComplexDish* dish) override {
            process((BasicDish*) dish);
            std::cout << "including:" << std::endl;
            for (auto sub_dish: dish->dishList()) {
                process((BasicDish*) sub_dish);
            }
        }
};

int main() {
    IDish* meat = new BasicDish("chicken", 200, 200);
    IDish* veg = new BasicDish("potato", 100, 50);
    ComplexDish* dinner = new ComplexDish("fried chicken & potatoes");
    dinner->addDish(meat);
```

```
    dinner->addDish(veg);

    IKitchenTool* pan = new FryingPan;
    IKitchenTool* analyzer = new DishAnalyzer;

    meat->apply(analyzer);
    veg->apply(analyzer);
    veg->apply(pan);
    dinner->apply(pan);
    dinner->apply(analyzer);
}
```

Завдання 5.

Для заданої предметної області спроектувати концептуальну модель бази даних (БД), яка повинна знаходитись у третій нормальній формі. В БД повинно бути щонайменше 5 таблиць. Представити два SQL-запити, один з яких охоплює не менше трьох таблиць. Представити очікувані відповіді.

Предметна область: Хімчистка

Хімчистка здійснює прийом у населення речей для виведення плям. Для наведення порядку складається база даних клієнтів, де зберігаються їх анкетні дані (прізвище, ім'я, по батькові, адресу, телефон). Починаючи з 3-го звернення, клієнт переходить у категорію постійних клієнтів і отримує знижку в 5% під час чищення кожної наступної речі. Всі послуги поділяються на види, що мають назву, тип і вартість, що залежить від складності робіт. Робота з клієнтом спочатку полягає у визначенні обсягу робіт, різновиду послуги та, відповідно, її вартості. Якщо клієнт погоджується, він залишає річ (при цьому фіксується послуга, клієнт та дата прийому) та забирає її після обробки (при цьому фіксується дата повернення). Хімчистка укладає із клієнтом договір. Клієнт може одночасно здавати в чищення декілька речей. У хімчистки з'явилися філії, і потрібна окрема статистика з філій. Введені надбавки за терміновість та складність.

ПРИКІНЦЕВІ ПОЛОЖЕННЯ

При виконанні завдань фахового іспиту **забороняється** використовувати будь-які інші допоміжні матеріали та електронні засоби (мобільні телефони, ноутбуки, планшети, тощо).

Критерії оцінювання (за системою ECTS, 100-бальна шкала)

Розв’язання кожної задачі оцінюється за такими критеріями:

95–100 – задачу розв’язано повністю, вірно;

85–94 – задачу розв’язано вірно, відповідь правильна, але наявними є один-два недоліки (наявними є деякі методичні помилки, порушено послідовність викладок тощо);

75–84 – задачу розв’язано вірно, але відповідь неправильна (наявними є арифметичні помилки);

65–74 – задачу розв’язано неповністю, але намічено правильний хід розв’язування;

60–64 – задачу не розв’язано, але наведено формули або твердження, що можуть бути використані при розв’язуванні задачі;

менше 60 – задачу не розв’язано.

Результат виконання фахового іспиту обчислюється як середнє арифметичне оцінок, отриманих за кожну задачу, і заокруглюється до цілих.

Згідно чинних «Правил прийому до КПІ ім. Ігоря Сікорського в 2024 році» при обчисленні конкурсного балу використовується шкала оцінювання від 100 до 200 балів. Перерахунок загального показника фахового іспиту у рейтингову оцінку фахового іспиту здійснюється згідно з Таблицею відповідності оцінок.

Таблиця відповідності оцінок РСО (60…100 балів)
оцінкам 200-бальної шкали (100…200 балів)

шкала РСО	шкала 100…200		шкала РСО	шкала 100…200		шкала РСО	шкала 100…200		шкала РСО	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. Зайченко О.Ю. Дослідження операцій Зайченко О.Ю., Зайченко Ю.П. — К: Видавничій дім «Слово», 2014. — 472 с.
2. Навчально-методичний посібник до практичних занять з курсу «Математичні методи оптимізації» для студентів магістратури усіх спеціальностей / Уклад. О.Ю.Зайченко. — К.: Політехніка, 2007. — 88 с.
3. Ладогубець В.В. Алгоритми параметричної оптимізації складних систем / Ладогубець В.В., Ладогубець Т.С., Ладогубець О.В. – К.: АВЕРС, 2006. – 139 с.
4. Фельдман Л.П. Чисельні методи в інформатиці /Петренко А.І., Дмитрієва О.А. — К.: Видавнича група BHV, 2006. — 480 с.
5. Задачин В.М. Чисельні методи: навч. пос. /В.М.Задачин, І.Г.Конюшенко. — Х.: ХНЕУ ім. С. Кузнеця, 2014. — 180 с.
6. Васильєв О. Програмування на C++ в прикладах і задачах. – К.: “Ліра – К”. – 2017. – 382 с. – ISBN 9786177507412.
7. Stroustrup B. A Tour of C++ (C++ In Depth Series, 3rd ed.). – Addison-Wesley Professional. – 2022. – 320 p. – ISBN 9780136816485.
8. Фрімен Е., Робсон Е., Бейтс Б., Сієрра К. Head First. Патерни проєктування. – Х.:“Фабула”. – 2020. – 672 с. – ISBN 9786170961594

9. Gamma E., Helm R., Johnson R., Vlissides J. Design Patterns: Elements of Reusable Object-Oriented Software. – Addison-Wesley. – 1995. – 395 p. – ISBN 0201633612

10. Booch G. Object-Oriented Analysis and Design with Applications. – Addison-Wesley Professional. – 2007. – 720 p. – ISBN 9780201895513

11. Шеховцов В. А. Операційні системи : підруч. для студ. / В. А. Шеховцов. — К. : Вид. група BHV, 2008. — 576 с.

12. Tanenbaum A. Modern Operating Systems, 4th ed. / Tanenbaum A., Bos H. — Pearson, 2014. — 1136 p.

13. Операційні системи: [Електронний ресурс]: навч. посіб. для студ. спеціальності 123 «Комп'ютерна інженерія» / В. Г. Зайцев, І. П. Дробязко; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 3 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2019. – 240 с.

14. Бондаренко М. Ф. Операційні системи : навч. посібник для вищих навч. закладів за напрямом "Комп'ютерні науки"/ М. Ф. Бондаренко, О. Г. Качко. — Х.: Компанія СМІТ, 2008. — 432 с.

15. Arnold K., Gosling J., Holmes D. The Java Programming Language (4th Edition). Addison-Wesley Professional, 2005. — 928 p.

16. Blanchette J., Summerfield M. C++ GUI Programming with Qt 4 (2nd Edition). Prentice Hall, 2008. — 752 p.

17. Васильєв О. Програмування мовою Java.- Видавництво Навчальна книга Богдан, 2020. - 696 с. ISBN: 9789661058797

18. Мартін Р. Чистий код. Створення і рефакторинг за допомогою Agile. - Харків: Видавництво Фабула, 2019. - 448 с. ISBN 978-617-09-5285-1

19. Мартін Р. Чиста архітектура. Мистецтво створення програмного забезпечення. Видання друге. - Харків: Видавництво Фабула, 2019. - 368 с. ISBN 978-617-09-5286-8

20. Exploring Cross-Platform Development with Flutter, React Native, and Xamarin by Eric Windmill. January 2020. Publisher(s): Manning Publications. - 98 pages.

21. Сегеда І.В. Проектування та використання баз даних-1. Комп'ютерний

практикум: навчальний посібник. КПІ ім. Ігоря Сікорського, 2021. 49 с. [Електронний ресурс]. – Режим доступу: https://ela.kpi.ua/handle/123456789/45902

22. Сегеда І.В., Дацюк О.А. Системи баз даних: Комп'ютерний практикум: навчальний посібник. КПІ ім. Ігоря Сікорського, 2019. 43 с. [Електронний ресурс]. – Режим доступу: https://ela.kpi.ua/handle/123456789/27252

РОЗРОБНИКИ ПРОГРАМИ

к.т.н., доцент кафедри СП НН ІПСА

Богдан БУЛАХ

к.т.н., доцент кафедри СП НН ІПСА

Олександр БЕЗНОСИК

к.т.н., доцент кафедри ММСА НН ІПСА

Ірина ШУБЕНКОВА

к.т.н., доц., доцент кафедри ЦТЕ НН ІАТЕ

Світлана ШАПОВАЛОВА

к.е.н., доц., доцент кафедри ЦТЕ НН ІАТЕ

Ірина СЕГЕДА

к.т.н., доц., в.о. зав. кафедри БМК ФБМІ

Світлана АЛХІМОВА

Програму рекомендовано:
кафедрою біомедичної кібернетики,
протокол №11 від 14 лютого 2024 року

в.о. зав. кафедри Світлана АЛХІМОВА