

Міністерство освіти і науки України
Хмельницький національний університет

«Затверджую»
Проректор з НІР
_____ Матюх С.А.
«__» _____ 2020р.

ПРОГРАМА
фахового вступного іспиту

з галузі знань 12 – Інформаційні технології
спеціальності 121 – Інженерія програмного забезпечення
освітня програма (спеціалізація) – «Інженерія програмного забезпечення»
для абітурієнтів — випускників коледжів та технікумів
(освітньо-професійної програми молодшого спеціаліста)

Затверджено на засіданні кафедри інженерії програмного забезпечення
Протокол № від _____ 2020р.

Завідувач кафедри ІІЗ
_____ д.ф.-м.н., проф. Бедратюк Л.П.

Затверджую
Декан ФПКТС
_____ к.т.н., проф. Савенко О.С.

Схвалено Вченою радою ФПКТС
Протокол №__ від __ _____ 2020р.

Голова Вченої ради ФПКТС
_____ к.т.н., проф. Савенко О.С.

Загальні положення

Вступні випробування проводяться приймальною комісією Хмельницького національного університету – за спеціальністю 121 - «Інженерія програмного забезпечення».

Під час виконання завдання перевіряються знання, вміння та навички студентів щодо розв'язання певних завдань з дисциплін «Основи інженерії програмного забезпечення», «Операційні системи», «Бази даних», «Об'єктно-орієнтоване програмування».

Критерії оцінювання

При тестуванні абітурієнт одержує завдання з 25 тестових питань – відповідно на перевірку основних знань і вмінь з дисциплін «Основи інженерії програмного забезпечення», «Операційні системи», «Бази даних», «Об'єктно-орієнтоване програмування».

Оцінюється тестове завдання за 200-бальною шкалою (від 100 до 200 балів), причому кожне тестове питання оцінюється рівною кількістю балів. Для подальшої участі в конкурсі абітурієнт повинен набрати не менше 124 балів.

Зміст навчального матеріалу

У програмі наведені в довільному порядку можливі теми, за якими сформовані тестові питання.

Основи програмування

Етапи розв'язування задач на ЕОМ. Алгоритм, його властивості та способи представлення. Базові алгоритмічні структури. Поняття програми, мови програмування, транслятора, системи програмування. Тестування та відладка програми. Загальний огляд C/C++. Синтаксис C. Типи даних. Змінні та константи. Стандартні математичні функції. Операції та вирази. Перетворення типів. Структура C-програми. Середовище розробки Microsoft Visual Studio. Організація консольного введення та виведення даних в C/C++. Форматоване введення/виведення даних. Функції printf() та scanf(). Введення/виведення символів та рядків. Введення/виведення у стилі C++. Основні правила конструювання лінійної програми. Оператори слідування. Складений оператор (блок). Прийняття рішень (розгалуження). Логічні вирази. Оператори, призначені для організації розгалужень (if, if...else, goto, switch, break). Арифметичні та ітераційні цикли. Оператори циклу for, while, do...while. Вкладені цикли. Оператори break та continue. Типові приклади. Поняття масиву у мові C. Одновимірні масиви. Оголошення та ініціалізація.

Формування, опрацювання та виведення одновимірних масивів. Наслідки виходу за межі діапазону зміни індекса. Організація двовимірних та багатовимірних масивів. Оголошення та ініціалізація. Опрацювання двовимірних масивів. Опрацювання багатовимірних масивів. Символьні масиви (рядки). Оголошення та ініціалізація. Введення, опрацювання та виведення символьних масивів. Бібліотечні функції мови С для роботи з рядками. Масиви рядків. Змінні, адреси та вказівники. Оголошення та ініціалізація вказівників. Операції над вказівниками. Вказівники та масиви. Вказівники та рядки. Масиви вказівників. Багаторівнева адресація. Способи виділення пам'яті для об'єктів у мові С. Динамічна пам'ять. Функції динамічного розподілу пам'яті. Моделювання одновимірних масивів з використанням динамічної пам'яті. Динамічне виділення пам'яті для багатовимірних масивів. Опрацювання динамічних масивів. Способи створення похідних типів даних у мові С. Тип перерахування. Переіменовування типів (typedef). Структури (struct). Оголошення та ініціалізація структур. Операції зі структурами. Масиви структур. Вкладені структури. Об'єднання (union). Бітові поля. Функції у мові С. Оголошення (опис) та визначення функції. Обмін даними між функціями. Оператор return. Параметри функцій: формальні, фактичні параметри, локальні та глобальні змінні. Виклик функції. Способи передачі даних у функцію (за значенням, за адресою). Функції зі змінним числом параметрів. Функції і вказівники. Масиви як параметри функцій. Програмування функцій, аргументами яких можуть бути вказівники. Функції, які повертають значення через вказівник. Вказівники на функції. Опрацювання структур у функціях. Способи передачі структур у функції, Створення функцій, які повертають структури і вказівники на структури. Рекурсивні алгоритми та функції. Поняття рекурсії. Види рекурсії та застосування рекурсивних алгоритмів. Програмування мовою С з використанням рекурсивних функцій. Модульна організація С-програм. Поняття модуля і модульного програмування. Структура складної С-програми. Міжфайлова взаємодія. Заголовні файли (.h-файли) і файли реалізації (.c-файли). Директиви препроцесора. Стандартні бібліотеки (.lib). Файлове введення/виведення. Поняття потоку і файлу в С. Текстові та бінарні файли. Файлова система ANSI C. Функції файлового введення/виведення. Вказівник на файл. Режими відкриття файлів. Перевірка на помилки відкриття файлів. Закриття файла. Запис/читання інформації. Форматоване введення/виведення. Довільний (прямий) доступ до файлів. Видалення файлів.

Основи інженерії програмного забезпечення

Програма як формалізований опис процесу обробки даних. Програмний засіб, програмний продукт, програмне забезпечення (ПЗ). Види ПЗ за різними ознаками. Складові вартості ПЗ. інженерія ПЗ як сукупність інженерних методів і засобів створення ПЗ. Предмет, зміст та практичне застосування предмету інженерії ПЗ. Інженери та інші фахівці в інженерії ПЗ. Стандартизація в інженерії ПЗ. Ядро знань Software Engineering Body of

Knowledge (SWEBOOK). Загальна характеристика областей знань SWEBOOK. Поняття життєвого циклу (ЖЦ) ПЗ. Основні, допоміжні та організаційні процеси ЖЦ ПЗ. Зв'язок між процесами. Класичні моделі ЖЦ (каскадна, спіральна, ітераційна, інкрементна), їх переваги та недоліки. Вибір моделі ЖЦ. Стандарти ЖЦ ПЗ. Методології швидкої розробки ПЗ. Гнучкі методології. RUP, XP, Scrum, сімейство методологій Crystal, DSDM (Dynamic Systems Development Method), FDD (Feature Driven Development), адаптивна розробка (ASD), методології MSF (Microsoft Solution Framework) та MOF (Microsoft Operations Framework). Місце гнучких методологій розробки ПЗ серед сучасних технологій розробки ПЗ. Поняття системи, моделі, моделювання. Основні підходи до аналізу і моделювання в інженерії ПЗ (структурний, об'єктно-орієнтований). Сутність та основні принципи структурного (функціонального) аналізу та моделювання. Основні положення методології Structured Analysis and Design Technique (SADT). Стандарти функціонального моделювання IDEF0 та IDEF3. Загальна методика побудови SADT-діаграм. Case-засоби автоматизації функціонального моделювання. Поняття діаграми потоків даних (Data Flow Diagrams, DFD). Синтаксис і семантика DFD-діаграм. Нотації Йордана-Де Марко і Гейна-Сарсона. Декомпозиція і відповідні розширення діаграм потоків даних. Побудова DFD-діаграм. Побудова ієрархії потоків даних. Автоматизація побудови DFD-моделей. Порівняльний аналіз SADT- та DFDмоделей. Поняття предметної області. Інфологічне моделювання та його мета. Концептуальний, логічний та фізичний рівні моделювання. Модель "сутність-зв'язок" (Entity-Relationship Diagrams, ERD). Конструктивні елементи інфологічних моделей. Фундаментальні типи зв'язків. Нормалізація даних. Огляд ER-нотацій. Загальна методика побудови ER-діаграм. Casesзасоби автоматизації інфологічного моделювання. Програма All Fusion Data Modeller (ERwin). Діаграми варіантів використання. Загальні поняття та сутність об'єктно-орієнтованого підходу (ООП) до розробки ПЗ. Принципи ООП. Методологія RUP (Rational Unified Process). Характеристика та концептуальні основи UML (Unified Modeling Language). Типи діаграм UML. Case-засоби автоматизації об'єктно-орієнтованого моделювання. Реалізація мови UML у середовищі IBM Rational Rose та MS Visual Studio. Побудова діаграм варіантів використання. Уніфікована мова моделювання UML. Деталізація варіантів використання. Способи опису варіантів використання. Потoki подій (основний, альтернативні). Текстовий опис потоків подій. Діаграми станів та діаграми діяльності, їх конструктивні елементи. Особливості побудови діаграм станів та діяльності. Уніфікована мова моделювання UML. Документування сценаріїв використання за допомогою діаграм взаємодії. Діаграми класів. Поняття сценарію використання. Основний та альтернативні сценарії. Форми представлення діаграм взаємодії. Діаграма послідовності та діаграма кооперації, їх елементи. Особливості побудови діаграм взаємодії. Діаграма класів, її елементи. Особливості побудови діаграм класів. Процес розробки вимог до ПЗ. Поняття вимоги до ПЗ. Класифікація вимог. Методології і стандарти, що

регламентують роботу з вимогами до ПЗ. Властивості вимог до ПЗ. Учасники розробки вимог до ПЗ. Загальний процес розробки вимог до ПЗ. Джерела вимог до ПЗ. Основні стратегії виявлення вимог. Особливості виявлення вимог до програмного продукту «під замовлення» і для відкритого ринку. Аналіз здійсності. Прототипування. Моделювання. Комунікації проектною командою і замовника в процесі розробки вимог до ПЗ. Мета аналізу вимог до ПЗ. Основні процедури верифікації та валідації вимог до ПЗ. Специфікація вимог до ПЗ. Специфікація функціональних вимог до ПЗ за допомогою варіантів використання. Формалізація вимог. Технічне завдання (ТЗ) на програмний продукт. Архітектура програмних засобів. Поняття структури та архітектури ПЗ. Мета архітектури. Основні класи архітектур. Архітектурні функції. Архітектурне представлення моделі проектування. Основні принципи проектування архітектури ПЗ. Архітектурні шаблони (патерни) та стилі. Методика побудови архітектури і дизайну. Графічне представлення архітектури. Проектування архітектури ПЗ на платформі Microsoft .NET. Аналіз та контроль архітектури. Фізичне представлення архітектурних моделей систем у мові UML (діаграми компонентів, діаграми розгортання). Базові концепції проектування ПЗ. Проектування ПЗ за SWEBOOK. Проектування ПЗ за стандартом ISO/IEC. Об'єкти проектування. Архітектурне та детальне проектування. Шаблони проектування. Методична, технологічна, інструментальна та організаційна підтримка процесу проектування ПЗ. Структурно-функціональне проектування ПЗ. Проектування ПЗ при об'єктному підході. Проектування інтерфейсу користувача. Основи конструювання ПЗ. Конструювання ПЗ за SWEBOOK. Управління конструюванням ПЗ. Моделі конструювання. Вимірювання в конструюванні. Огляд основних парадигм програмування. Класифікація мов програмування. Стилі та методи програмування. Інструментальні засоби та інтегровані середовища програмування. Транслятори. Кодування. Правила оформлення модулів. Інтеграція. Оптимізація програм. Способи економії пам'яті. Способи скорочення часу виконання. Стандарти конструювання ПЗ. Поняття та мета тестування ПЗ. Основні види тестування. Рівні тестування. Стратегії тестування. Локалізація та виправлення програмних помилок. Особливості тестування об'єктно-орієнтованих модулів та об'єктно-орієнтованої інтеграції. Тестування інтерфейсу користувача. Верифікація та валідація (атестація) ПЗ. Розробка планів тестування. Стратегія розробки тестових наборів даних. Основні поняття в області якості ПЗ. Метрики і моделі якості програмних систем. Стандарти якості ПЗ. Управління конфігурацією в життєвому циклі ПЗ. Впровадження, супровід та моніторинг ПЗ. Процеси управління конфігурацією програмних засобів. Етапи і процедури при управлінні конфігурацією ПЗ. Технологічне забезпечення при управлінні конфігурацією ПЗ. Основні етапи готовності програмного продукту (альфа-версія, бета-версія, реліз). Організація, методи і процедури супроводу ПЗ. Техніки супроводу. Реінжинірінг, зворотній реінжинірінг. Рефакторинг. Реверсна інженерія. Задачі і процеси перенесення програм і даних на інші платформи. Загальні принципи розробки документації,

пов'язаної з життєвим циклом ПЗ. Передпроектна та проектна документація. Документація, призначена для користувача ПЗ. Електронна документація. Інтерактивні електронні керівництва. Стандарти, що регламентують розробку документації ПЗ. Основні документи Єдиної системи програмної документації (ЄСПД) та Єдиної системи конструкторської документації (ЄСКД). Технічні письменники. Автоматизація розробки програмної документації.

Операційні системи

Огляд комп'ютерних пристроїв та операційних систем. Класифікація комп'ютерних систем. Визначення поняття "операційна система" (ОС). Огляд функціональності і призначення ОС. Основні режими роботи користувачів і завдань в ОС (пакетний, мультипрограмування, поділ часу). Огляд особливостей ОС для різних класів обчислювальних пристроїв (багатопроцесорні і розподілені системи, настільні, кишенькові, мобільні та ін.). ОС реального часу, ОС для хмарних обчислень. Специфіка вимог до ОС і архітектур ОС для розглянутих класів пристроїв. Архітектура і функціональність ОС. Архітектура ОС і її функціональність; управління процесами як основна функція ОС; огляд базових механізмів синхронізації процесів - семафорів і моніторів. Огляд функціональності ОС: управління пам'яттю, файлами, процесами, мережами, командними інтерпретаторами, сервіси ОС, системні виклики; організація ОС за принципом рівнів абстракції. Методи проектування та реалізації ОС: рівні абстракції ОС; ОС з архітектурою мікроядра; віртуальні машини; мети проектування і розробки ОС з точки зору користувачів і розробників; генерація ОС для конкретної комп'ютерної системи при її інсталяції. Процеси та потоки. Основні концепції управління процесами, планування та диспетчеризації процесів. Поняття процесу. Стану процесу. Блок управління процесом. Перемикання з одного процесу на інший. Черги, пов'язані з диспетчеризацією процесів. Планувальники, що виконують диспетчеризацію процесів. Перемикання контексту. Створення і знищення процесу. Взаємодія процесів: проблема обмеженого буфера; проблема "виробник - споживач"; прямий і непрямий зв'язок процесів; клієнт-серверний взаємозв'язок; сокетний зв'язок; віддалений виклик процедури (RPC) і віддалений виклик методу (RMI); вибудовування параметрів (marshaling). Поняття потоку (thread) і багатопоточне виконання (multi-threading); моделі багатопоточності; користувальницькі потоки і потоки ядра; потоки в "Ельбрусі", Solaris, Linux, POSIX, Windows 2000, Java. Диспетчеризація та синхронізація процесів. Тупики. Планування і диспетчеризація процесора; критерії диспетчеризації; стратегії диспетчеризації (FCFS, SJF, RR); багаторівневі черги, диспетчеризація мультипроцесорних систем і систем реального часу. Синхронізація процесів: критичні секції; алгоритми розв'язання

проблеми взаємного виключення критичних секцій; двійкові і загальні семафори; рішення проблем "обмежений буфер", "читачі-письменники", "філософи, що обідають"; монітори; синхронізація в linux і Windows. Поняття тупика (deadlock), модель системи, граф розподілу ресурсів, граф wait-for, методи обробки та запобігання тупиків. Поняття безпечного стану системи, алгоритм банкіра, алгоритми виявлення тупиків. Керування пам'яттю. Принципи управління пам'яттю; пристрій управління пам'яттю; логічне і фізичне адресні простори; динамічна лінковка; оверлейна структура програми. Сторінкова організація пам'яті. Відкачування і підкачування (swapping); стратегії динамічного розподілу пам'яті; фрагментація; принципи сторінкової організації; таблиця сторінок; використання асоціативної пам'яті; дворівневі, ієрархічні, хешировані і інвертовані таблиці сторінок; поділювані сторінки. Сегментна організація пам'яті; сегментно-сторінкова організація пам'яті. Принципи сегментної організації пам'яті. Архітектура сегментної організації пам'яті. Приклад сегментної організації пам'яті. Приклад використання поділюваних сегментів. Сегментно-сторінкова організація пам'яті в системах MULTICS і "Ельбрус". Сегментно-сторінкова організація пам'яті в системі Intel386. Віртуальна пам'ять. Концепція віртуальної пам'яті; сторінкова організація віртуальної пам'яті; обробка відсутності сторінки в пам'яті (pagefault); обробка сторінок на вимогу; спільне використання сторінок процесами; файли, які відображаються в пам'ять (memory-mappedfiles); стратегії заміщення сторінок; алгоритми FIFO і LRU; алгоритм "другого шансу"; алгоритми з лічильником; виділення фреймів - фіксоване і з пріоритетами; thrashing; сторінкова організація в WindowsNT і Solaris. Системи файлів. Інтерфейс із системою файлів; структура файлу і операції над файлом; типи файлів; відмінності файлових систем в ОС MULTICS і "Ельбрус" від файлових систем в ОС для ПЕОМ; файли послідовного і прямого доступу; директорія, способи організації директорій; монтування файлових систем; захист файлів; блок управління файлом; принципи реалізації файлових систем. Віртуальні файлові системи (VFS); реалізація файлів за допомогою FAT (Windows) та індексних блоків (UNIX); управління зовнішньою пам'яттю; кешування; файлові системи на основі журналу транзакцій; мережева файлова система NFS. Системи введення-виведення. Організація введення-виведення в комп'ютерній системі та її підтримка в ОС; контролери; драйвери; контролери з прямим доступом до пам'яті (DirectMemoryAccess- DMA); цикл виконання завдання в ОС, чергування обчислень і введення-виведення, переривання.

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.

роздяднула WHERE. Порядок виконання оператора SELECT. Реалізація реляційної алгебри засобами оператора SELECT (Реляційна повнота SQL. Оператор декартового добутку. Оператор проєкції. 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, неакуратне 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.

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 й 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 С #. 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 (закон для розробника). Другий закон (закон для користувача). Третій закон (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.

Розділ 2

Перелік екзаменаційних питань

Бази даних, Об'єктно-орієнтоване програмування

1. Основні етапи розв'язування задач за допомогою ЕОМ.
2. Базові алгоритмічні структури.
3. Транслятор. Види трансляторів, їх принципи роботи, достоїнства та недоліки.
4. Алфавіт та лексеми мови С.
5. Стандартні типи даних мови С.
6. Змінні та константи мови С. Оголошення та ініціалізація.
7. Стандартні математичні функції мови С.
8. Вирази та арифметичні операції мови С. Чим відрізняються префіксне і постфіксне інкрементування і декрементування?
9. Перетворення типів у виразах. Неявне і явне приведення типів.
10. Загальна структура С-програми. Яке значення повертає головна функція main() проекту у С-програмах?
11. Загальна схема підготовки виконуваної програми (.exe-файла). Які основні засоби для цього необхідні?
12. 12. Інтегроване середовище розробки Microsoft Visual Studio. Рішення і проект. У якому вікні Microsoft Visual Studio міститься графічне представлення всього рішення?
13. Форматоване введення даних. Функція scanf(). Основні специфікатори формату.
14. Форматоване виведення даних. Функція printf(). Основні специфікатори формату.
15. Введення/виведення символів. Функції getchar() і putchar().
16. Введення/виведення рядків. Функції gets() і puts().
17. Локалізація консольних додатків у мові С.
18. Охарактеризуйте оператори слідування в лінійній програмі.
19. Логічні операції та вирази.
20. Охарактеризуйте умовний оператор if. Для чого призначений оператор goto?
21. Охарактеризуйте оператор вибору switch.
22. Охарактеризуйте тернарний оператор ?:.
23. Синтаксис та порядок виконання циклу for.
24. Синтаксис та порядок виконання циклу while. Яку мінімальну кількість разів може виконатись цикл while?
25. Синтаксис та порядок виконання циклу do...while. Яку мінімальну кількість разів може виконатись цикл do...while?

26. Дострокове завершення циклу. Оператори break та continue. У чому відмінність і подібність між цими операторами?
27. Одновимірні масиви у мові C. Оголошення та ініціалізація.
28. Формування, опрацювання та виведення одновимірних масивів. На кого чи на що покладається контроль границь числових масивів?
29. Двовимірні масиви у мові C. Оголошення та ініціалізація. Яка мінімальна кількість циклів потрібна для роботи з двовимірними масивами?
30. Формування, опрацювання та виведення двовимірних масивів. На кого чи на що покладається контроль границь числових масивів?
31. Символьні масиви (рядки). Оголошення та ініціалізація. Що таке константа EOF і з якою метою вона використовується?
32. Введення/виведення символьних масивів (рядків). Особливості введення рядків за допомогою функції scanf().
33. Основні бібліотечні функції мови C для роботи з рядками.
34. Масиви рядків. Оголошення та ініціалізація.
35. Змінні, адреси та вказівники. Оголошення та ініціалізація вказівників. Який сенс має значення вказівника NULL?
36. Основні операції над вказівниками.
37. Вказівники та одновимірні масиви.
38. Вказівники та рядки.
39. Вказівники та багатовимірні масиви.
40. Масиви вказівників, оголошення та ініціалізація. Багаторівнева адресація.
41. Способи виділення пам'яті для об'єктів у мові C. Динамічний розподіл пам'яті.
42. Функції динамічного розподілу пам'яті у мові C. Яка різниця у дії функцій malloc() і calloc()?
43. Моделювання одновимірних масивів з використанням динамічної пам'яті.
44. Динамічне виділення пам'яті для багатовимірних масивів (1-й спосіб).
45. Динамічне виділення пам'яті для багатовимірних масивів (2-й спосіб).
46. Вільні масиви. Особливості виділення динамічної пам'яті під вільний масив.
47. Охарактеризуйте тип даних "перерахування".
48. Поняття структури у мові C. Оголошення та ініціалізація. У чому полягає відмінність структури від масиву?
49. Операції зі структурами та з елементами структур.
50. Масиви структур та елементи-масиви.
51. Вкладені структури.
52. Об'єднання. Оголошення та ініціалізація.
53. Функції у мові C. Оголошення (опис) та визначення функції.

54. Параметри функцій (глобальні, локальні, формальні та фактичні). Виклик функції.
55. Загальні правила організації функцій у мові C.
56. Способи передачі даних у функцію. Оператор return.
57. Функції зі змінним числом параметрів.
58. Одновимірні масиви як параметри функцій. Чи може функція за допомогою return повернути масив?
59. Двовимірні масиви як параметри функцій.
60. Функції, які повертають значення через вказівник.
61. Опрацювання структур у функціях.
62. Охарактеризуйте рекурсивні функції.
63. Охарактеризуйте суть модульного програмування.
64. Призначення препроцесора мови C. Основні директиви препроцесора.
65. Заголовні файли та їх основні елементи.
66. Специфікатори класів пам'яті при оголошенні змінних у багатофайловій програмі.
67. Область видимості змінних.
68. Потоки і файли у мові C. Стандартні потоки. Текстові та бінарні файли.
69. Файлова система ANSI C. Загальна характеристика.
70. Вказівник на файл. Відкриття та закриття файла. Як у мові C кодується ознака кінця файла?
71. Запис у файл та читання з файла символів. Функції getc() і fgetc().
72. Запис у файл та читання з файла рядків. Функції fprintf() і fscanf().
73. Синтаксис та призначення функцій feof() та rewind() при роботі з файлами.
74. Читання/запис блоків даних. Функції fread() і fwrite().
75. Довільний (прямий) доступ до файлів і функція fseek().
76. Поняття програми, програмного засобу, програмного забезпечення. Види ПЗ.
77. Причини та передумови виникнення дисципліни «Інженерія ПЗ». Визначення інженерії ПЗ.
78. Основні області знань SWEBOOK.
79. Життєвий цикл програмного забезпечення.
80. Загальна характеристика каскадної моделі життєвого циклу ПЗ; її переваги і недоліки.
81. Загальна характеристика спіральної моделі життєвого циклу ПЗ та її переваги і недоліки.
82. Загальна характеристика ітераційної моделі життєвого циклу ПЗ та її переваги і недоліки.
83. Загальна характеристика інкрементної (покрокової) моделі життєвого циклу ПЗ та її переваги і недоліки.
84. Загальна характеристика методології швидкої розробки додатків RAD (Rapid Application Development) та її переваги і недоліки.
85. Гучкі методології розробки ПЗ.

86. Міжнародні, міждержавні та національні стандарти життєвого циклу ПЗ.
87. Сутність та методи структурного аналізу і моделювання.
88. Загальна характеристика методологій структурного аналізу і моделювання.
89. Основні принципи структурного аналізу і моделювання.
90. Нотація IDEF0 для графічного представлення SADT-моделі.
91. Синтаксис і семантика DFD-діаграм.
92. Базові поняття інформаційного моделювання «сутність-зв'язок» (ERD).
93. Огляд нотацій, що використовуються при побудові ER-діаграм.
94. Методика побудови ER-діаграм.
95. Сутність об'єктно-орієнтованого аналізу і моделювання; його переваги і недоліки.
96. Основні принципи об'єктно-орієнтованого аналізу і моделювання.
97. Загальна характеристика мови UML.
98. Діаграми UML.
99. Процес розробки вимог до ПЗ.
100. Джерела вимог до ПЗ.
101. Стратегія виявлення вимог до ПЗ.
102. Аналіз, систематизація та перевірка вимог до ПЗ.
103. Специфікація та формалізація вимог до ПЗ.
104. Технічне завдання на ПЗ (по ГОСТ 19.201-78).
105. Поняття архітектури ПЗ. Основні класи архітектур.
106. Основні принципи проектування архітектури ПЗ.
107. Модульна структура програмного продукту. Основні характеристики модулів. Типи модулів та їх структура.
108. Низхідне («зверху-донизу») та висхідне («знизу-доверху») проектування ПЗ.
109. Розробка структури ПЗ при об'єктному підході.
110. Принципи, правила та етапи проектування інтерфейсу ПЗ.
111. Класифікації діалогів і загальні принципи їх розробки. Реалізація діалогів в графічному інтерфейсі користувача.
112. Парадигми програмування.
113. Поняття мови програмування. Етапи розвитку мов програмування.
114. Класифікація мов програмування.
115. Інструментальні засоби та інтегровані середовища програмування. Транслятори.
116. Основи конструювання ПЗ. Стандарти в конструюванні. Управління конструюванням.
117. Поняття технології програмування. Структурні та об'єктно-орієнтовані технології програмування.
118. Класифікація програмних помилок. Методи відладки ПЗ. Загальна методика відладки ПЗ.
119. Основні види тестування.

120. Рівні тестування (модульне, інтеграційне та системне тестування).
121. Тести. Класифікація тестів. Валідаційні тести. Вимірювання результатів тестування.
122. Особливості об'єктно-орієнтованого тестування.
123. Особливості тестування об'єктно-орієнтованих модулів.
124. Верифікація та атестація ПЗ.
125. Процеси управління конфігурацією програмних засобів.
126. Етапи і процедури при управлінні конфігурацією ПЗ.
127. Технологічне забезпечення при управлінні конфігурацією ПЗ.
128. Основні етапи готовності програмного продукту. Альфа-версія. Бета-версія. Реліз.
129. Організація і методи супроводу ПЗ. Етапи і процедури при супроводі ПЗ.
130. Техніки супроводу. Реінжинірінг, зворотній реінжинірінг.
131. Задачі і процеси перенесення програм і даних на інші платформи.
132. Загальні принципи складання та оформлення документації ПЗ.
133. Характеристика документації, призначеної для користувача ПЗ.
134. Електронна довідкова система.
135. Основні стандарти на програмну документацію.
136. Огляд комп'ютерних пристроїв та операційних систем
137. Архітектура і функціональність ОС
138. Методи проектування та реалізації ОС
139. Процеси та потоки
140. Диспетчеризація та синхронізація процесів. Тупики
141. Керування пам'яттю
142. Файлові системи
143. Системи введення-виведення
144. Основні етапи і процеси адміністрування інформаційних систем та мереж.
145. Необхідність захисту інформаційних систем і телекомунікацій.
146. Інформаційні мережі, їх використання при розподіленій обробці.
147. Основи маршрутизації в IP-мережах.
148. Захист даних в мережах.
149. Мережеві функції операційних систем сімейства MS Windows, їх особливості.
150. Функції і призначення команди NET.
151. Мережеві і персональні операційні системи (ОС).
152. Служби каталогів, функції та призначення.
153. Управління об'єктами ActiveDirectory.
154. Методи забезпечення безпеки аутентифікації користувачів в розподілених системах, схема Kerberos.
155. Управління доступом до об'єктів системи, списки прав доступу (ACL).
156. Файловий сервер.
157. Шаблони безпеки.

158. Стратегія і методика адміністрування.
159. Автоматизація адміністративних завдань.
160. Зберігання даних в інформаційних системах.
161. Технології об'єктно-орієнтованого проектування програмних систем
162. Особливості створення додатків в .NetFramework та особливості мови С#
163. Класи та об'єкти С#
164. Абстрактні класи. Структури та перерахування
165. Концепція наслідування
166. Інтерфейси.
167. Множинне наслідування
168. Делегати.
169. Події .
170. Універсальність. Класи з родовими параметрами
171. Коректність і стійкість програмних систем.
172. Бази даних. Функції СКБД.
173. Загальні поняття реляційного підходу до організації БД.
174. Базові поняття реляційної моделі даних.
175. Реляційна алгебра.
176. Мови запитів до баз даних.
177. СКБД MySQL
178. СКБД Access(Open Office.org Base).
179. Нормальні форми відношень.
180. Нормальні форми більш високих порядків.
181. Транзакції і цілісність баз даних.
182. СКБД в архітектурі "клієнт-сервер".
183. Розробка мережеских баз даних.
184. Розподілені бази даних.
185. Системи керування базами даних наступного покоління.

Перелік джерел посилання

1. Лавріщева К.М. Програмна інженерія: підручник. Київ: Академперіодика, 2008. 415с.
2. Радельчук Г. І. Основи програмної інженерії: Лабораторний практикум для студентів напряму підготовки “Програмна інженерія”. Хмельницький: ХНУ, 2012. 108 с.
3. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++. 2-е изд.: Пер. с англ. - Москва: Издательство Бином, 2007. 560 с.
4. Бородкіна І.Л. Бородкіна Г.О. Інженерія програмного забезпечення: посібник для студентів вищих навчальних закладів: навч. посіб. Київ: Центр навчальної літератури (ЦНЛ), 2019. 204 с.
5. Леоненков А.В. Нотация и семантика языка UML. - Електронний ресурс. - <http://www.intuit.ru/department/pl/umlbasics/>
6. Бабич А. В. Введение в UML. - Електронний ресурс. - <http://www.intuit.ru/department/se/intuml/>
7. Д.В.Кознов Визуальное моделирование: теория и практика. - Електронний ресурс. - <http://www.intuit.ru/department/se/vismodtp/>
8. Бабич А. В. Введение в UML. – Електронний ресурс. – <http://www.intuit.ru/department/se/intuml/>
9. Браунде Э. Технология разработки программного обеспечения. Санкт-Петербург : Питер., 2011. – 655 с.
10. Орлов С.А. Технологии разработки программного обеспечения. (Разработка сложных программных систем): учеб. пособие. Санкт-Петербург : Питер. 2012. 464с.
11. Карпенко М. Ю., Манакова Н. О., Гавриленко І. О. Технології створення програмних продуктів та інформаційних систем: навч. посібник. Харків: Нац. ун-т міськ. госп-ва ім. О. М. Бекетова, 2017. 93 с.
12. Мацяшек Л.А. Анализ требований и проектирование систем. Разработка информационных систем с использованием UML. пер. с англ. Москва: Вильямс. 2012. 432с.
13. Томашевський В.М. Моделювання систем: Підручник. Київ: Видавнича група ВНУ. 2015. 352с.
14. Мінухін С.В., Беседовський О.М., Знахур С.В. Методи і моделі проектування на основі сучасних CASE-засобів: навч. посіб. Харків: ХНЕУ. 2010.-272с.
15. Форкун Ю.В. Методичні вказівки до виконання курсового проекту для студентів напряму підготовки “Програмна інженерія” з дисципліни «Архітектура та проектування програмного забезпечення. Хмельницький: ХНУ, 2015. – 46 с.
16. ДСТУ ISO/IEC 2382:2017 (ISO/IEC 2382:2015, IDT). Інформаційні технології. Словник термінів. Частина 15. Мови програмування (ISO/IEC 2382:2015, IDT). Вид. офіц. Київ: Держстандарт України, 2019.

17. Інформатика: Комп'ютерна техніка. Комп'ютерні технології. Посіб. За ред. О.І.Пушкаря. Київ: Видавничий центр "Академія", 2012. 696с.
18. Макконнелл С. Совершенный код. Мастер-класс. пер. с англ. Москва: Издательско-торговый дом "Русская Редакция", 2011. 896.
19. Дудзяний І.М. Об'єктно-орієнтоване моделювання програмних систем: Навчальний посібник. Львів: Видавничий центр ЛНУ імені Івана Франка, 2010. 108 с.
20. Ларман, Крэг. Применение UML и шаблонов проектирования. 2-е издание.: пер. с англ. Москва: Вильямс, 2011. 624 с.
21. Мещанінов О. П. Моделювання систем: Навчальний посібник. - Миколаїв: Вид-во МФ НаУКМА, 2011. 268с.
22. Мінухін С.В. Case-технології [Текст]: Конспект лекцій. Харків: ХНЕУ.- 2015. 156с.
23. Орлов С. Технологии разработки программного обеспечения: Учебник. Санкт-Петербург, Питер, 2012. 464 с.
24. Гибкая методология разработки программного обеспечения. Санкт-Петербург: Русская редакция, 2011. 127 с
25. Резник С., Бьерк А., де ла Маза М. Scrum с Team Foundation Server 2010. Профессиональный подход. Москва: ЭКОМ Паблишерз, 2012.
26. Майкл С.В. Тернер Основы Microsoft Solutions Framework. Санкт-Петербург: Русская редакция, 2013. 336 с.
27. Бази даних. Методичні вказівки до курсового проектування для студентів напряму підготовки "Програмна інженерія". Хмельницький : ХНУ, 2014. – 28 с.
28. Кузнецов С. Основы баз данных : учебное пособие. Москва: Интернет-Университет Информационных Технологий, 2011. 484 с.
29. Гольцман В. MySQL 6.0. Санкт-Петербург: Питер, 2015. 255 с.

Зразок базового тестового завдання

1. Яка діаграма UML моделює представлення функціонального призначення системи?

Виберіть одну відповідь:

- a. діаграма варіантів використання
- b. діаграма компонентів
- c. діаграма станів
- d. діаграма класів
- e. інша відповідь

2. Охарактеризуйте метод висхідного проектування програмного забезпечення:

Виберіть одну відповідь:

- a. це методика детального проектування програмного забезпечення, визначена у стандартах ЄСПД
- b. це методика розробки програм "знизу догори", яка починається з розробки підпрограм (процедур, функцій), у той час, коли опрацьовування загальної архітектури системи, що розробляється, ще не закінчилося
- c. це методика розробки програм, при якій розробка починається з визначення цілей вирішення проблеми, після чого йде послідовна деталізація, що закінчується детальною програмою
- d. це методика розробки програм, при якій модульна деревовидна структура програми формується у процесі програмування модуля
- e. інша відповідь

3. Модель програмної системи з певної точки зору, в якій відображені лише суттєві аспекти і опущено все, що є несуттєвим при даному погляді на систему, - це:

Виберіть одну відповідь:

- a. архітектурний шаблон
- b. архітектурне представлення (View)
- c. архітектурний стиль
- d. модельний стиль
- e. інша відповідь

4. Атрибут або група атрибутів, що однозначно ідентифікує екземпляр сутності в ER-моделі - це ...

Виберіть одну відповідь:

- a. зовнішній ключ
- b. альтернативний ключ
- c. первинний ключ
- d. простий ключ
- e. інша відповідь

5. Що таке прототип функції?

Виберіть одну відповідь:

- a. опис функції, включаючи її тип, ім'я, типи і обов'язкові імена параметрів
- b. опис функції, включаючи її ім'я, тип, імена і типи параметрів, тіло функції
- c. опис функції, включаючи її тип та ім'я
- d. опис функції, включаючи її тип, ім'я і типи параметрів
- e. інша відповідь

6. Під матрицю a , що містить n рядків і m стовпців, виділена динамічна пам'ять у вигляді "неперервної стрічки". Вкажіть вірне звернення до елементів такої матриці (у загальному вигляді):

Виберіть одну відповідь:

- a. $*(a+i*m+j)$
- b. $*(a[i]+j)$
- c. $*(a+i*n+j)$
- d. $a[i][j]$
- e. інша відповідь

7. Яким буде значення змінної k після виконання фрагмента C-програми:
`int k=10; ++k;`

Виберіть одну відповідь:

- a. 11
- b. 9
- c. 10
- d. 20
- e. інша відповідь

8. Що буде надруковано в результаті виконання наступного фрагмента C-програми: `char s[10] = "test"; printf("%d",strlen(s));`

Виберіть одну відповідь:

- a. 10
- b. 4
- c. test
- d. нічого
- e. інша відповідь

9. У C-програмі оголошена структура `tovar`. Вкажіть вірне оголошення змінної a типу структури:

Виберіть одну відповідь:

- a. `struct tovar a;`
- b. `tovar struct a;`
- c. `tovar a;`
- d. `void struct tovar a;`
- e. інша відповідь

10. У якому випадку оператор return в тілі функції є необов'язковим?

Виберіть одну відповідь:

- a. оператор return в тілі функції є обов'язковим завжди
- b. якщо тип функції не вказаний
- c. якщо функція має тип void
- d. є необов'язковим
- e. інша відповідь

11. Використовуючи змінну a, вкажіть вірне оголошення для вказівника на вказівник на цілочисельне значення:

Виберіть одну відповідь:

- a. int *a;
- b. int **a;
- c. *int *a;
- d. int a;
- e. інша відповідь

12. Призначення конструктора як функції-члена класу полягає в тому, що ...

- a. Конструктор копіює об'єкти різних класів
- b. Конструктор викликається автоматично під час оголошення класу
- c. Конструктор повертає значення даних-членів класу з методів, що змінюють значення цих даних-членів
- d. Конструктор створює об'єкт класу
- e. Інша відповідь

13. Об'єктно-орієнтоване програмування — це

- a. Методологія програмування, яка базується на поданні програми у вигляді сукупності процедур
- b. Інша відповідь
- c. Методологія програмування, яка базується на поданні програми у вигляді сукупності модулів
- d. Методологія програмування, яка базується на поданні програми у вигляді сукупності класів, які утворюють ієрархію, на принципах спадкування
- e. Методологія програмування, яка базується на поданні програми у вигляді сукупності процедур і функцій

14. Абстракція – це:

- a. Інша відповідь
- b. Характеристика змінних, які використовуються в процедурах
- c. Різновид процедур в програмі
- d. Різновид процедур та функцій
- e. Характеристики змінних, які використовуються в функціях

15. Модульність – це:

- a. Розбиття програми на функції
- b. Розбиття програми на процедури та функції

- c. Розбиття програми на окремі класи
- d. Розбиття програми на процедури
- e. Інша відповідь

16. Наслідування – це:

- a. Відношення між модулями, коли один модуль може використовувати можливості іншого
- b. Відношення між змінними, коли значення однієї змінної також є значенням іншої c. Інша відповідь
- d. Відношення між функціями, коли функціональні можливості однієї функції може використовувати інша
- e. Відношення між ділянками оперативної пам'яті, коли одну ділянку пам'яті можуть використовувати різні змінні

17. Типізація – це:

- a. Інша відповідь
- b. Характеристика змінних, які використовуються в модулях
- c. Різновид абстракції, який дозволяє використовувати змішані класи
- d. Обмеження, яке накладається на функцію і перешкоджає змішуванню різних функцій
- e. Обмеження, яке накладається на модуль і перешкоджає змішуванню різних модулів

18. Операція створення і (або) ініціалізації об'єкта — це:

- a. Деструктор
- b. Інша відповідь
- c. Ітератор
- d. Модифікатор
- e. Селектор

19. Операція знищення об'єкта — це:

- a. Селектор
- b. Конструктор
- c. Ітератор
- d. Модифікатор
- e. Інша відповідь

20. Операція, яка надає доступ для визначення стану об'єкта без його змін— це:

- a. Ітератор
- b. Конструктор
- c. Інша відповідь
- d. Модифікатор
- e. Деструктор

21. Вираз DELETE FROM КАСА означає:

- a. Видалення окремого рядка
- b. Видалення атрибутів таблиці
- c. Видалення усіх рядків таблиці
- d. Видалення таблиці
- e. Інша відповідь

22. Вираз SELECT Назва AS Назва_Організації... означає:

- a. Імені стовпця для вибору даних
- b. Перевизначення імені таблиці
- c. Перевизначення імені стовпця
- d. Перевизначення імені таблиці
- e. Інша відповідь

23. Віртуальні таблиці зберігають:

- a. Текст запиту SELECT, і дані
- b. Тільки дані результату виконання запиту SELECT
- c. Тільки текст запиту SELECT
- d. Тільки індекси і хеш-таблиці
- e. Інша відповідь

24. Реалізація системних викликів: використовуючи асемблер, програміст встановлює значення регістрів і / або областей пам'яті, а потім виконує спеціальну інструкцію виклику сервісу або програмного переривання для звернення до деякої

- a. Програми
- b. Підпрограми
- c. Підсистеми ос
- d. Функції ос
- e. Інша відповідь

25. В ОС, кожен процес яких має тільки один потік, виникають проблеми організації паралельних обчислень в рамках

- a. Поточку
- b. Процесу
- c. Функції
- d. Переривань
- e. Інша відповідь

Зразок додаткового завдання

1. Функції динамічного розподілу пам'яті у мові С. Яка різниця у дії функцій malloc() і calloc()?
2. Управління об'єктами ActiveDirectory.
3. Множинне наслідування.