

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

Затверджую

Голова Приймальної комісії
Ректор



Михайло ЗГУРОВСЬКИЙ

25.04.2024р
дата

ПРОГРАМА
додатково вступного випробування

для вступу на освітньо-наукову програму підготовки доктора філософії
«Інженерія програмного забезпечення»

за спеціальністю 121 Інженерія програмного забезпечення

Програму ухвалено:

Науково-методичною комісією за спеціальністю
121 Інженерія програмного забезпечення

Протокол № від «__» «_____» 2024 р.

Голова НМК

Євгенія СУЛЕМА

Зміст

1. Загальні відомості
2. Питання, що виносяться на вступне випробування
3. Навчально-методичні матеріали
4. Рейтингова система оцінювання
5. Приклад екзаменаційного білету

1. Загальні відомості

Ця Програма призначена для вступників на навчання для здобуття ступеня доктора філософії за спеціальністю 121 Інженерія програмного забезпечення інформації (освітня програма «Інженерія програмного забезпечення»).

Метою Програми є надання вступникам інформації щодо змісту навчального матеріалу, перевірка знань з якого відбувається під час додаткового фахового вступного випробування для вступу на навчання для здобуття ступеня доктора філософії за спеціальністю 121 Інженерія програмного забезпечення (освітня програма «Інженерія програмного забезпечення»).

Вступне випробування зі спеціальності проводиться у формі письмового екзамену, який включає п'ять тестових завдань з варіантами відповідей. Для складання вступного випробування надається 60 хвилин.

Зміст навчального матеріалу, перевірка знань з якого відбувається під час фахового вступного випробування, наведений у розділі 2 «Питання, що виносяться на вступне випробування» цієї Програми.

Для підготовки до випробування рекомендовано використовувати навчальну літературу, наведену у розділі 3 «Навчально-методичні матеріали».

Інформація щодо методики оцінювання наведена у розділі 4 «Рейтингова система оцінювання». Приклад білету наведено у розділі 5 «Приклад екзаменаційного білету».

2. Питання, що виносяться на вступне випробування

1. Загальні положення та основні поняття математичного моделювання. Класифікація моделей. Властивості та контроль правильності математичних моделей. δ -функція Дірака та функція Хевісайда. Елементи теорії похибок. Технологія моделювання.
2. Математичні моделі, що описуються диференціальними рівняннями першого порядку. Загальні відомості про диференціальні рівняння та методи їх розв'язання. Точки рівноваги розв'язків диференціального рівняння та їх стійкість.
3. Математичні моделі, що описуються диференціальними рівняннями вищих порядків. Лінійні диференціальні рівняння вищих порядків. Механічні коливання.
4. Математичні моделі, що описуються системами диференціальних рівнянь. Системи диференціальних рівнянь. Задачі, які зводяться до розв'язання систем диференціальних рівнянь
5. Аналіз систем диференціальних рівнянь та моделей, що описуються системами диференціальних рівнянь. Лінеаризація нелінійних систем, аналіз їх фазових портретів та стійкості. Моделі взаємодії популяцій.
6. Комбінаторна оптимізація. Обчислювальна складність. Підходи до розв'язання.
7. Алгоритми з заборонами. Загальна ідея побудови алгоритмів з заборонами. Алгоритм з заборонами для розв'язання задачі розфарбування графа. Алгоритм з заборонами для розв'язання задачі комівояжера. Алгоритм з заборонами для розв'язання квадратичної задачі про призначення.
8. Мурашині алгоритми. Загальна ідея побудови мурашиних алгоритмів. Варіанти побудови мурашиних алгоритмів: AS, EAS, MMAX, ASrank. Мурашиний алгоритм для розв'язання задачі знаходження найкоротшого шляху між двома вершинами графа. Мурашиний алгоритм для розв'язання задачі квадратичних призначень. Мурашиний алгоритм для розв'язання задачі розфарбування графа. Мурашиний алгоритм для розв'язання задачі комівояжера.
9. Алгоритми кластеризації. Задача кластеризації. Типи кластеризації. Алгоритм K-means. Алгоритм Fuzzy C-means. Алгоритм кластеризації множини відповідно до заданого критерію.
10. Штучні нейронні мережі. Штучний нейрон. Класифікація штучних нейронних мереж. Одно- й багатошаровий перцептрон. Алгоритм навчання. Мережа зворотного поширення (Back Propagation). Алгоритм навчання. Мережа зустрічного поширення (Counter Propagation).

- Алгоритм навчання. Карта Кохонена. Алгоритм навчання. Мережа Хопфилда. Алгоритм навчання. Двоскерована асоціативна пам'ять.
11. Елементи класичної теорії оптимізації. Основні положення. Необхідні та достатні умови безумовного екстремуму. Необхідні та достатні умови умовного екстремуму
 12. Лінійне програмування. Постановка задачі та основні положення. Графічний метод розв'язання задач лінійного програмування. Симплекс-метод Данцига. Елементи теорії двоїстості. Метод оберненої матриці. Двоїстий симплекс-метод
 13. Дискретне програмування. Метод Гоморі (метод відсікаючих площин). Метод гілок та границь.
 14. Задача про найкоротші шляхи для всіх пар. Динамічне програмування для знаходження найкоротшого шляху. Алгоритм Флойда-Воршелла. Алгоритм Джонсона.
 15. Постановка завдання кластеризації. Функції відстані. Критерії якості кластеризації. Алгоритм К-середніх та модифікації.
 16. Алгоритми пошуку найкоротшого шляху. Задача про найкоротші шляхи з одного входу. Пошук дерева найкоротшого шляху. Алгоритм Дійкстри.
 17. Розподілені системи. Синхронізація часу. Алгоритм Крістіана. Алгоритм Берклі. Транзакція. Принципи АСІД. Архітектура MapReduce.
 18. Нормальний розподіл. Крива Гауса. Функція розподілу нормального закону. Правило «трьох сігм».
 19. Функція декількох випадкових величин. Розподіл суми і добутку незалежних випадкових величин. Числові характеристики випадкових величин: математичне сподівання, дисперсія.
 20. Конкурентні системи. Паралелізм. Моделі конкурентного програмування. Shared memory. Модель обміну повідомленнями. Проблеми гонки за ресурси (racescondition).
 21. Характеристика моделей життєвого циклу програмного продукту: каскадна, спіральна, ітеративна, інкрементна (пошагова), формальна, швидкого прототипування. Їх переваги, недоліки та використання.
 22. Визначення та задачі аналізу вимог до програмного забезпечення. Вимоги до продукту та процесу. Функціональні та нефункціональні вимоги. Вимоги з кількісною оцінкою. Системні та програмні вимоги.
 23. Структура та архітектура програмного забезпечення. Архітектурні точки зору та представлення. Характеристика архітектур програмних продуктів. Поняття шаблонів проектування, фреймворки.
 24. Загальні концепції проектування програмного забезпечення. Концепції та процес проектування програмного забезпечення. Загальні принципи (техніки) проектування.

25. Фундаментальні питання та проблеми проектування програмного забезпечення. Паралелізм та ефективна обробка програмних процесів.
26. Контроль та обробка подій. Розподіл компонентів. Обробка помилок та виключних ситуацій. Взаємодія системи з користувачами та збереження даних.
27. Моделювання у програмній інженерії. Основні принципи моделювання програмних систем. Візуальне моделювання та CASE-технології. Уніфікована мова моделювання UML. Призначення UML. Види діаграм UML. Діаграми структури та діаграми поведінки. Спрощена стратегія використання UML-діаграм при моделюванні програмного забезпечення.
28. Діаграми прецедентів та їх використання. Моделювання контексту та вимог до програмної системи. Прецеденти та потоки подій. Описи потоків подій. Потоки подій та сценарії як типи й екземпляри.
29. Специфікація прецедентів. Використання сценаріїв при плануванні версій. Зв'язки між акторами та прецедентами. Організація прецедентів (відношення залежності, включення та розширення). Відношення узагальнення для прецедентів та акторів. Реалізація прецедентів та використання діаграм послідовностей.
30. Побудова діаграм послідовності. Двохетапне розроблення діаграм послідовностей. Діаграми співробітництва.
31. Узгодженість (цілісність) моделей. Виявлення класів. Класи етапу аналізу: прикордонні (boundary) або інтерфейсні класи; класи-сутності (entity); управляючі (control) класи (класи-менеджери). Класи етапу проектування. Відношення між класами (узагальнення, залежність, асоціація, агрегація, композиція) та їх виявлення. Проектування класів, відношень між класами. Пакетування класів.
32. Класифікація патернів: породжуючі патерни (патерни процесу створення об'єктів); структурні патерни (що ґрунтуються на композиціях – структурних об'єднаннях об'єктів чи класів); патерни поведінки (що характеризуються взаємодією об'єктів між собою). Успадкування та композиції. Простір патернів проектування. Приклади патернів. Кодогенерація (інженірінг) та реінженірінг.
33. Стратегії та методи проектування програмного забезпечення. Загальні стратегії. Функціонально-орієнтоване та структурне проектування. Об'єктно-орієнтоване проектування. Проектування на основі структур даних. Компонентне проектування.
34. Будова програми при використанні процедурного (імперативного) програмування. Проблеми, що виникають при використанні процедурного програмування та їх вирішення використанням об'єктно-орієнтованого програмування.

35. Характеристика структурного програмування як методології розробки програмного забезпечення. Базові конструкції програми. Поняття підпрограми (процедури або функції). Метод розробки програми. Мета та принципи модульного програмування.
36. Методології розробки програмного забезпечення Microsoft Solution Framework, Rational Unified Process, Extreme Programming. Засоби розробки, що базуються на них.
37. Мета та задачі супроводу програмного забезпечення. Еволюція програмного забезпечення та потреба у його супроводі. Категорії та ключові питання супроводу програмного забезпечення.
38. Мета та задачі управління програмними проектами. Аналіз здійсненності програмного проекту: технічні, операційні, фінансові та соціальні аспекти. Процес оцінки та перегляду вимог до програмного забезпечення. Планування програмного проекту. Управління виконанням програмного проекту.
39. Основні принципи конструювання програмного забезпечення: мінімізація складності, очікування змін, конструювання з можливістю перевірки, стандарти конструювання. Загальні принципи керування конструюванням програмного забезпечення. Планування та вимірювання в конструюванні програмного забезпечення. Мови конструювання та програмне кодування. Тестування та якість конструювання. Загальні аспекти повторного використання.
40. Поняття якості програмного забезпечення. Показники якості розробки програмного забезпечення. Аналіз якості та техніки оцінювання. Виміри в проектуванні програмного забезпечення. Характеристика основних критеріїв якості програмного продукту. Основні принципи та задачі керування якістю програмного забезпечення. Характеристика дефектів та методи управління якістю програмного забезпечення.
41. Термінологія та ключові питання тестування програмного забезпечення. Зв'язок тестування з іншими видами діяльності процесу розробки програмного забезпечення. Мета та рівні тестування. Техніки тестування та їх комбінація. Вимір результатів тестування.
42. Структура програм на мовах програмування Python, C/C++ або Java. Прості та складені оператори, приклади використання операторів. Поняття константи та змінної в мовах програмування, принципи виділення пам'яті для розміщення констант та змінних. Оголошення та ініціалізація констант і змінних у програмах на мовах програмування Python, C/C++ або Java. Прості типи даних. Розміщення даних різних типів в оперативній пам'яті. Приклади оголошення різних типів даних у програмах на мовах програмування Python, C/C++ або Java. Сумісність типів та приведення типів даних. Типи даних користувача. Оголошення та використання таких типів в мовах програмування Python, C/C++ або

Java. Операції, вирази та оператори в мовах програмування Python, C/C++ або Java. Операції привласнення та арифметичні операції. Пріоритети операцій. Зміна пріоритетів.

43. Логічні операції та умовні оператори. Алгоритмічна структура розгалуження. Вибір із двох альтернатив. Блок схеми структур розгалуження. Вкладеність конструкцій вибору. Алгоритмічні конструкції повторення. Типи циклів. Блок схеми циклів. Організація циклів. Поняття функції (підпрограми) в мовах програмування. Оголошення та виклик функцій. Рекурсія функцій. Стандартні процедури та функції. Поняття глобальних та локальних змінних та приклади їх використання в програмах на мовах Python, C/C++ або Java. Статичні та динамічні змінні. Поняття масиву та його властивості, їх оголошення і використання. Одновимірні та багатовимірні масиви. Доступ до елементів масиву. Рядки. Операції над символами та рядками в мовах програмування Python, C/C++ або Java. Стандартні процедури та функції обробки рядків. Записи та структури, їх оголошення і використання. Доступ до компонентів та операції над записами та структурами. Похідні типи даних. Основні концепції об'єктно-орієнтованої методології програмування. Програмна модель об'єкту. Наведіть приклад програмної моделі довільного об'єкту на мові програмування Python, C++ або Java. Оголошення класів на мові програмування Python, C++ або Java.
44. Конструктори і деструктори класу. Призначення і основні правила використання. Наведіть приклад оголошення довільного класу та його використання в тексті програм на мові програмування. Структура програми, що використовує класи, на мові програмування Python, C++ або Java. Наведіть приклади.
45. Поняття інкапсуляції та її застосування в мовах програмування. Методи які організують інтерфейсі. Наведіть приклади визначення відкритих закритих полів і методів класу. Поняття спадкування та поліморфізму в об'єктно-орієнтованих мовах програмування. Поняття та приклади перевантажених функцій. Ієрархія класів. Одиночне і множинне спадкоємство. Абстрактні класи. Призначення і використання. Наведіть приклад абстрактного класу. Спеціальні методи класів. Перевантаження операторів. Поняття дружніх функцій і класів. Особливість доступу до закритих полів класу. Наведіть приклади організації доступу до закритих полів.
46. Організація обмінних операцій з використанням бібліотеки класів для введення - виведення. Поняття файлу і потоку, приклади роботи з ними. Маніпулятори потоків.
47. Поняття шаблону функції та класу. Абстрактні алгоритми.
48. Бібліотека стандартних шаблонів STL (Standard Template Library).

- 49.Компілятори і інтерпретатори. Опис їх роботи та порівняння переваг і недоліків. Основні задачі, що вирішуються компілятором. Вхідні і вихідні файли, їх логічний зміст.
- 50.Поняття про ресурси ЕОМ. Операційна система як програма управління ресурсами. Основні завдання управління програмними процесами, оперативною пам'яттю і пристроями введення - висновку в операційній системі.
- 51.Управління процесами в багатозадачних ОС. Основні алгоритми управління.
- 52.Управління оперативної пам'яттю в багатозадачній ОС. Основні алгоритми управління. Віртуальна пам'ять.
- 53.Управління ресурсами. Монопольне та спільне використання ресурсів. Поняття моделювання ресурсу. «Спулінг».
- 54.Критичні секції в асинхронних процесах багатозадачних ОС.
- 55.Основні алгоритми синхронізації. Завдання синхронізації процесів. Поняття критичного шляху. «Змагання» і «клінчі».
- 56.Основні методи організації зберігання файлів на зовнішніх пристроях прямого методу доступу. Поняття бібліотечної, індексної послідовної і кластерної структури.
- 57.Типи діалогів, порівняльні характеристики діалогів. Групи користувачів ОС. Основні характеристики діалогів типу меню та команда.
- 58.Групи користувачів ОС.
- 59.Способи досягнення надійності програмного забезпечення.
- 60.Поняття про тестування. Навести простий приклад тестування.
- 61.Мова запитань до баз даних SQL. Особливості застосування. Альтернативні мови запитання до баз даних. Особливості застосування.
- 62.Методологія розробки програмного забезпечення. Технологічний процес створення програмного забезпечення
- 63.Моделі життєвого циклу розробки програмного забезпечення. Водоспадна модель. Моделі життєвого циклу розробки програмного забезпечення. Спіральна модель. Раціональний уніфікований процес. Методологія розробки програмного забезпечення. Екстремальне програмування.
- 64.Загальна архітектура комп'ютера. Класифікація архітектур комп'ютерів за Флінном. Основні положення архітектури комп'ютера Фон Неймана.
- 65.Загальні принципи передачі файлів та повідомлень через комп'ютерні мережі, переваги пакетної передачі. Характеристика рівням моделі

взаємодії відкритих систем (Open System Interconnect - OSI), приклади протоколів різних рівнів.

66. Формат MAC-адреси, поняття однопунктових (Unicast), багатопунктових (Multicast) та широкомовних (Broadcast) MAC-адреси. Приклади використання протоколами кадрів із такими типами MAC-адрес. Поняття колізії при організації колективного доступу до середовища передачі в технології Ethernet. Алгоритм метода колективного доступу до середовища передачі з розпізнанням несучої та виявленням колізій (Carrier-Sense-Multiply-Access with Collision Detection – CSMA/CD). Загальна характеристика технології Ethernet. Успадковані технології Fast Ethernet, Gigabit Ethernet і 10 Gigabit Ethernet, їх спільні риси. Загальна характеристика протоколів мережного рівня. Формат IP-адреси та принцип виділення блоків IP-адрес за класовою схемою. Особливі IP-адреси та правила їх вживання. Планування IP-підмереж за допомогою масок підмереж. Відображення IP-адрес на адреси каналного рівня за допомогою протокола ARP. Модель роботи ARP. Відображення доменних імен на IP-адреси, етапи процесу визначення IP-адреси за доменним ім'ям. Основні задачі протоколу IP. Формат заголовку та характеристика полів, що вирішують задачі IP. Принципи маршрутизації IP-пакетів у складеній мережі, структура таблиць маршрутизації та команди задавання статичного маршруту до мереж. Загальна характеристика та основні задачі протоколу контрольних повідомлень Інтернету ICMP. Типи повідомлень ICMP. Принципи використання ICMP повідомлень в утилітах ping та traceroute. Задачі протоколів транспортного рівня. Порівняльна характеристика протоколів TCP та UDP. Поняття портів та адрес сокетів, розподіл номерів портів. Номери портів популярних протоколів прикладного рівня. Процедури встановлення з'єднання, передачі даних та завершення з'єднання протоколом TCP. Поля та прапорці заголовків пакетів, що використовуються при цих процедурах. Формат заголовку TCP. Використання полів та прапорців для цілісної та гарантованої передачі даних.
67. Поняття бази даних та системи управління базами даних (СУБД). Поняття сутності, атрибуту, зв'язку та домену та їх ілюстрація на довільному прикладі. Найбільш поширені СУБД. Порівняльна характеристика реляційної та об'єктно-орієнтованої моделей даних. Приклади СУБД, що використовують такі моделі.
68. Проблема надлишкового дублювання та узгодження введення дубльованих даних. Вимоги до вибору ідентифікаторів записів. Аномалії вставки, видалення та модифікації даних. Поняття цілісності даних. Базові типи зв'язків між даними. Формування зв'язків типів «один до одного», «один до багатьох» та «багато до багатьох». Приклади даних із такими типами зв'язків. Елементи моделі сутність-зв'язок. Основні

визначення ER-діаграм. Приклади ER-діаграм з різноманітними типами зв'язків.

69. Поняття нормалізації бази даних. Використання методу нормальних форм для оптимізації структури відношень. Залежності між атрибутами. Виявлення залежностей між атрибутами. Критерій першої нормальної форми, поняття та приклади повторюваних груп та способу їх усунення. Критерій другої нормальної форми, поняття та приклади функціональної залежності атрибутів та визначника (детермінанта). Критерій третьої нормальної форми, поняття та приклади транзитивної залежності між атрибутами відношення. Критерій нормальної форми Бойса-Кодда.
70. Команди маніпуляції із таблицями: команда додавання стовпців, команда модифікації характеристик стовпців, команда перейменування та команда видалення стовпців, команда видалення таблиць. Синтаксис оператора SELECT. Приклади виразу SQL, що організує вибірку даних із двох таблиць для певного критерію. Синтаксис операторів INSERT, DELETE, UPDATE. Приклади виразів SQL, що організують вставку нових даних, видалення та модифікацію даних, відповідно. Використання вкладених підзапитів в мові SQL. Принцип роботи запиту зі зв'язаним підзапитом. Приклад організації запиту з підзапитом.
71. Оператор об'єднання UNION. Принцип роботи запиту при використанні оператора. Приклад використання оператора UNION. Усунення дублювання в UNION. Операції поєднання таблиць шляхом посиальної цілісності. Різновиди та принципи роботи оператора об'єднання таблиць JOIN. Приклади використання оператора JOIN.

3. Навчально-методичні матеріали

1. Дичка І.А., Онай М.В. Математичні моделі, що описуються диференціальними рівняннями та їх системами: методичні вказівки до виконання лабораторних робіт з дисципліни «Математичне моделювання систем та процесів» для студентів спеціальностей 8.05010301 «Програмне забезпечення систем» та 8.05010302 «Інженерія програмного забезпечення» [Електронне видання] / І. А. Дичка, М. В. Онай. – К. : НТУУ «КПІ», 2013. – 130 с. – Режим доступу: http://www.fpm.kpi.ua/archive/dir.do?sys_id=obj_12685
2. Згуровський М.З., Панкратова Н.Д. Основи системного аналізу. – К.: Видавнича група ВНУ, 2007. – 544 с.
3. Забара С.С., Гагарін О.О., Кузьменко І.М., Щербашин Ю.Д. Моделювання систем у середовищі MATLAB: Навч. посіб. К.: Університет «Україна», 2011. – 137 с.
4. Томашевский В.М. Моделювання систем. – К.: Видавнича група ВНУ, 2007. – 352 с.: іл.
5. Субботін С.О., Олійник А.О., Олійник О.О. Неітеративні, еволюційні та мультиагентні методи синтезу нечіткологічних і нейромережних моделей. Монографія / Під заг. ред. С.О.Субботіна. – Запоріжжя: ЗНТУ, 2009.- 375с.
6. Ю.П. Зайченко. Дослідження операцій. Підручник. Сьоме видання, перероблене та доповнене. – К.: Видавничий Дім —Слово, 2006. – 816 с.
7. Зайченко О.Ю., Зайченко Ю.П. Дослідження операцій. Збірник задач. – К.: Видавничий дім —Слово, 2007. – 472 с.
8. Горбань О.М. Основи теорії систем та системного аналізу [Текст]: / Горбань О.М., Бахрушин В.Є. – Запоріжжя: ГУ «ЗІДМ У», 2004.
9. Вінник В. Ю. Алгоритмічні мови та основи програмування. Мова С. – Житомир: ЖДТК, 2007. – 328 с.
10. Керніган Б., Річі Д. Мова програмування С, друге видання. – 232 с.
11. The Java™ Tutorials – офіційна документація. – Режим доступу до ресурсу. <https://docs.oracle.com/javase/tutorial>

4. Рейтингова система оцінювання

Вступне випробування проводять лише за затвердженим комплектом екзаменаційних білетів. Відмова студента від написання вступного випробування за екзаменаційним білетом атестується як незадовільна відповідь.

При виявленні факту використання недозволених матеріалів екзаменаційна комісія має право припинити випробування і виставити незадовільну оцінку.

Для написання вступного випробування надається не більше 60 хвилин. П'ять тестових завдань із вибором однієї правильної відповіді оцінюються у 20 балів за кожне. Сумарна кількість балів набраних вступником за фахове випробування складає 100 балів: $20+20+20+20+20=100$ балів.

Додаткове вступне випробування оцінюється за шкалою «зараховано», «незараховано». Випробування вважається складеним, якщо набрано 60 і більше балів. В цьому випадку вступник отримує «зараховано».

5. Приклад екзаменаційного білету

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Освітній ступінь – доктор філософії
Спеціальність 121 Інженерія програмного забезпечення

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № ___

Тестове завдання 1.

Дано код:

```
public static void main(String[] args) {  
    int x = -2, y = 5, max;  
    // додайте код  
    System.out.println(max);  
}
```

Який із запропонованих фрагментів виведе в консоль максимальне число?
(кілька правильних відповідей)

- if (x < y) max = x; else max = y;
- if (x > y) max = x; else max = y;
- max = x > y ? x : y;
- max = x < y ? x : y;

Тестове завдання 2.

Як викликати стандартний метод класу всередині static-метода?

- ніяк, static-метод не працює з об'єктом класу
- можна, треба перед цим перевантажити звичайний метод класу
- можна, треба перед цим перевизначити звичайний метод класу
- можна, нічого додатково робити не треба

Тестове завдання 3.

Валідація вимог – це перевірка викладених у специфікації вимог, яка виконується для того, щоб переконатися в релевантності вимог для програмної системи, що розробляється.

- А. Валідація вимог – це перевірка викладених у специфікації вимог, яка виконується для того, щоб переконатися в релевантності вимог для програмної системи, що розробляється.
- В. Валідація вимог – це документування викладених у специфікації вимог, яке виконується для того, щоб переконатися в релевантності вимог для програмної системи, що розробляється.
- С. Валідація вимог – це збирання і викладення у специфікації вимог, щоб погодити їх з замовниками
- D. Валідація вимог – це процес перевірки правильності специфікацій вимог щодо їх відповідності потребам, несуперечності, повноти і можливості реалізації, а також узгодженості зі стандартами.

Тестове завдання 4.

Для моделювання сценаріїв програмної системи використовується анотація

- A. IDEF3
- B. IDEF0
- C. DFD
- D. IDEF5

Тестове завдання 5.

Програмна інженерія відрізняється від традиційної промислової інженерії природою свого продукту,

- A. який не відчувається і не матеріалізується в наочний фізичний предмет
- B. який відчувається і не матеріалізується в наочний фізичний предмет
- C. який не відчувається і матеріалізується в наочний фізичний предмет
- D. який відчувається і матеріалізується в наочний фізичний предмет

Затверджено
Гарант освітньої програми

Євгенія СУЛЕМА

РОЗРОБНИКИ ПРОГРАМИ:

Сулема Євгенія Станіславівна – гарант освітньої програми, завідувач кафедри програмного забезпечення комп'ютерних систем, доктор технічних наук, доцент

Барабаш Олег Володимирович – професор кафедри інженерії програмного забезпечення в енергетиці, доктор технічних наук, професор

Клименко Ірина Анатоліївна – професор кафедри обчислювальної техніки, доктор технічних наук, професор

Новотарський Анатолій Михайлович – професор кафедри обчислювальної техніки, доктор технічних наук, професор

Люшенко Леся Анатоліївна – доцент кафедри програмного забезпечення комп'ютерних систем, кандидат технічних наук, доцент

Стіренко Сергій Григорович – завідувач кафедри обчислювальної техніки, доктор технічних наук, професор

Стеценко Інна Вячеславівна – професор кафедри інформатики та програмної інженерії, доктор технічних наук, професор