

1. Обробка запитів Web-сервером у асинхронному режимі.
2. Реалізація систем Web-сервісу, тенденції розвитку.
3. Хмарні обчислення та моделі хмарних інфраструктур.
4. Програмне забезпечення систем хмарних обчислень.
5. Апаратна віртуалізація систем хмарних обчислень.
6. Архітектура систем хмарних обчислень.
7. Структура Amazon Web Services.
8. Призначення, структура та характеристики Amazon S3.
9. Призначення, структура та характеристики Amazon EC2.
10. Концепція Amazon EC2.
11. Доступ до сервісу Amazon EC2.
12. Встановлення екземпляру Amazon EC2.
13. Доступ до екземпляру Amazon EC2.
14. Групи безпеки та зони доступу.
15. Зберігання даних в Amazon EC2.
16. Управління образами віртуальних машин АМІ.
17. Надійність сервісу хмарних обчислень.
18. Продуктивність сервісу хмарних обчислень.
19. Побудова образів віртуальних обчислювальних машин.
20. Масштабування хмарної інфраструктури.

Дисципліна «Теорія прийняття рішень»

Проф. Годлевський М.Д.

- 1 Метод обмежень при пошуку компромісних рішень в задачах векторної оптимізації.
- 2 Труднощі та проблеми унікального вибору.
- 3 Етапи прийняття рішень.
- 4 Типи шкал.
- 5 Постановки задач прийняття рішень.
- 6 Класифікація методів оцінювання та порівняння багатокритеріальних альтернатив.
- 7 Аксиоматичні методи оцінювання альтернатив.
- 8 Прямі методи оцінювання альтернатив.
- 9 Методи компенсацій. Методи порогів непорівняння.
- 10 Людино-машинні процедури прийняття рішень.
- 11 Поняття ефективних і слабо ефективних оцінок і рішень.
- 12 Теоретичне і практичне значення поняття ефективного рішення.
- 13 Метод обмежень при пошуку компромісних рішень в задачах векторної оптимізації.
- 14 Лексикографічні задачі. Приклади лексикографічних задач.
- 15 Лексикографічна задача про покриття.
- 16 Лексикографічна транспортна задача.
- 17 Задача про ранець.
- 18 Зображення лексикографічного відображення функціоналом.
- 19 Рішення багатокритеріальних задач методом поступлень.
- 20 Класифікація задач системної оптимізації.

1. Характерні особливості сучасного етапу розвитку в сфері розробки програмного забезпечення систем (*software system*) та інформаційних технологій (*information technology*): чинники впливу та якісні критерії успіху виконання ІТ-проектів.
2. Стисла порівняльна характеристика основних методологій проектування (*design methodology*) сучасних програмних систем (ПС). Виходячи з аналізу предметної задачі своєї дипломної роботи, мотивувати вибір певної методології проектування ПС та надати приклади проектних рішень (з використанням відповідної нотації), що мають бути розроблені при цьому.
3. RUP (*Rational Unified Process*) - методологія проектування програмних систем (ПС): загальна схема та стисла характеристика окремих етапів. Виходячи з аналізу предметної задачі своєї дипломної роботи, навести приклади проектних артефактів, що мають бути створені за методологію RUP при розробці відповідної ПС.
4. Особливості об'єктно - орієнтованого підходу (*object-oriented approach*) до проектування програмних систем (ПС). Загальна характеристика уніфікованої мови моделювання систем UML. Виходячи з аналізу предметної задачі своєї дипломної роботи, розглянути можливість і доцільність застосування цього підходу для проектування ПС та навести приклади окремих проектних рішень (UML - діаграм).
5. Методика розробки UML-діаграм для етапів накопичення вимог (*requirements engineering*) та виконання системного аналізу (*system analysis*) при застосуванні об'єктно-орієнтованого підходу в проектуванні програмних систем (ПС). Виходячи з аналізу предметної задачі своєї дипломної роботи навести приклади цих діаграм для ПС, що має бути розроблена.
6. Методика розробки UML-діаграм для етапу проектування програмного забезпечення (*software design*) при застосуванні об'єктно-орієнтованого підходу в проектуванні програмних систем (ПС). Виходячи з аналізу предметної задачі своєї дипломної роботи навести приклади цих діаграм для ПС, що має бути розроблена.
7. Методика розробки UML-діаграм для етапу програмної реалізації (*software construction*) при застосуванні об'єктно-орієнтованого підходу в проектуванні програмних систем (ПС). Виходячи з аналізу предметної задачі своєї дипломної роботи навести приклади цих діаграм для ПС, що має бути розроблена.
8. Метод проектування SADT (*Structured Analysis Design Technique*): загальна характеристика та сфери застосування при проектуванні програмних систем (ПС). Виходячи з аналізу предметної задачі своєї дипломної роботи, розглянути можливість і доцільність застосування цього методу для проектування ПС, або запропонувати альтернативний підхід та навести приклади проектних рішень з використанням відповідної нотації.

9. Структурно-функціональна методологія проектування програмних систем (ПС): DFD (Data Flow Diagrams) діаграми та ER (Entity-Relationship) моделі. Виходячи з аналізу предметної задачі своєї дипломної роботи, розглянути можливість і доцільність застосування цих методів для проектування ПС та навести приклади проектних рішень з використанням відповідної нотації.
10. Доменні онтології (*domain ontology*) в процесі проектування програмних систем (ПС): визначення поняття та методика побудови. Виходячи з аналізу предметної задачі своєї дипломної роботи, навести приклад побудови фрагменту доменної онтології з застосуванням нотації UML при розробці відповідної ПС.
11. Проблеми розробки архітектури програмної системи (ПС). Визначення поняття системної архітектури (*system architecture*) та класифікація типів системних архітектур. Виходячи з аналізу предметної задачі своєї дипломної роботи, мотивувати вибір певного типу системної архітектури при розробці відповідної ПС.
12. Архітектурні моделі типу «клієнт-сервер»: порівняльний огляд та технології їх побудови в програмних системах (ПС). Виходячи з аналізу предметної задачі своєї дипломної роботи, мотивувати доцільність вибору певного типу архітектури «клієнт-сервер» при розробці відповідної ПС.
13. Патерни (шаблони) проектування (*design pattern*) – визначення поняття, властивості та приклади. Характеристика патерна MVC. Виходячи з аналізу предметної задачі своєї дипломної роботи, навести приклади ПП, що можуть бути використані при розробці відповідної ПС.
14. Методи та технології проектування компонентних програмних рішень (КПР) в розподілених програмних системах (ПС).. Виходячи з аналізу предметної задачі своєї дипломної роботи, проаналізувати проблеми проектування КПР при розробці відповідної ПС.
15. Проблеми оцінки ефективності функціонування програмних рішень (*software performance engineering*) при проектуванні та експлуатації програмних системах (ПС). Класифікація методів моделювання компонентних програмних рішень (КПР). Виходячи з аналізу предметної задачі своєї дипломної роботи, навести приклади КПР, що мають бути досліджені щодо ефективності їх функціонування при розробці відповідної ПС.
16. Методика застосування імітаційного моделювання для дослідження функціонування компонентних програмних рішень (КПР) в програмних системах (ПС). Виходячи з аналізу предметної задачі своєї дипломної роботи, навести приклади КПР, що мають бути досліджені щодо ефективності їх функціонування при розробці відповідної ПС.
17. Методика застосування апарату метричних моделей (побудова графів виконання) для дослідження функціонування програмних рішень (КПР) в програмних системах (ПС). Виходячи з аналізу предметної задачі своєї дипломної роботи, навести приклади КПР, що мають бути досліджені щодо ефективності їх функціонування при розробці відповідної ПС.
18. Загальна характеристика проблем реінжинірингу успадкованих програмних системах (ПС) та класифікація методів проведення реінжинірингу. Виходячи

з аналізу предметної задачі своєї дипломної роботи, мотивувати необхідність або недоцільність застосування методів ре-інжинірингу при розробці відповідної ПС.

19. Проблеми та підходи до реінжинірингу ресурсів даних (*data resources*) в успадкованих програмних системах (ПС). Виходячи з аналізу предметної задачі своєї дипломної роботи, розглянути проблему реінжинірингу ресурсів даних при розробці відповідної ПС і запропонувати певне рішення, якщо це потрібно.
20. Проблеми та підходи до реінжинірингу програмної архітектури (*software architecture*) в успадкованих програмних системах (ПС). Виходячи з аналізу предметної задачі своєї дипломної роботи, розглянути проблему реінжинірингу програмної архітектури при розробці відповідної ПС і запропонувати відповідне рішення, якщо це потрібно.