Управління проектами

ДЕНЬ 4

Проектування програмної системи

Проектування

Процес визначення архітектури, компонентів, інтерфейсів та інших характеристик системи або її компонентів називається **проектуванням**

Результат процесу проектування - проект (design)

Види проектування

Концептуальне проектування

Логічне проектування

Фізичне проектування

Архітектура

Архітектура програмного забезпечення (software architecture) - опис підсистем і компонент програмної системи, а також зв'язків між ними

Архітектура визначає внутрішню структуру одержуваної системи, задаючи спосіб, яким система організована або конструюється

Різновиди архітектури систем

Монолітна

Клієнт-сервер

Тришарова

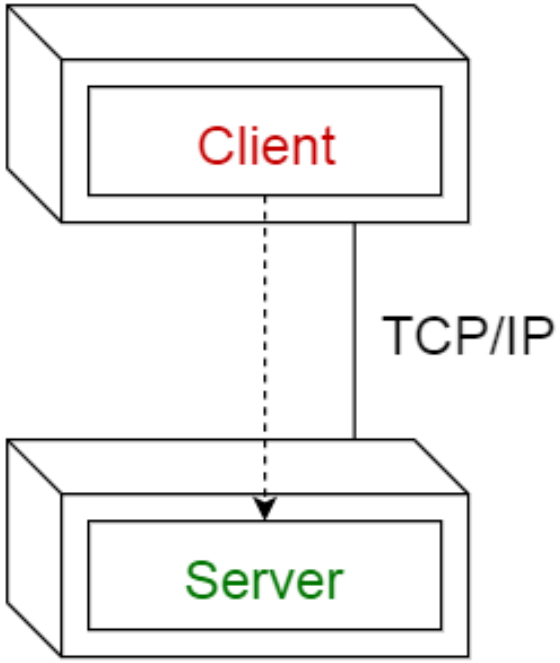
Багатошарова розподілена архітектура

Сервісно-орієнтована архітектура

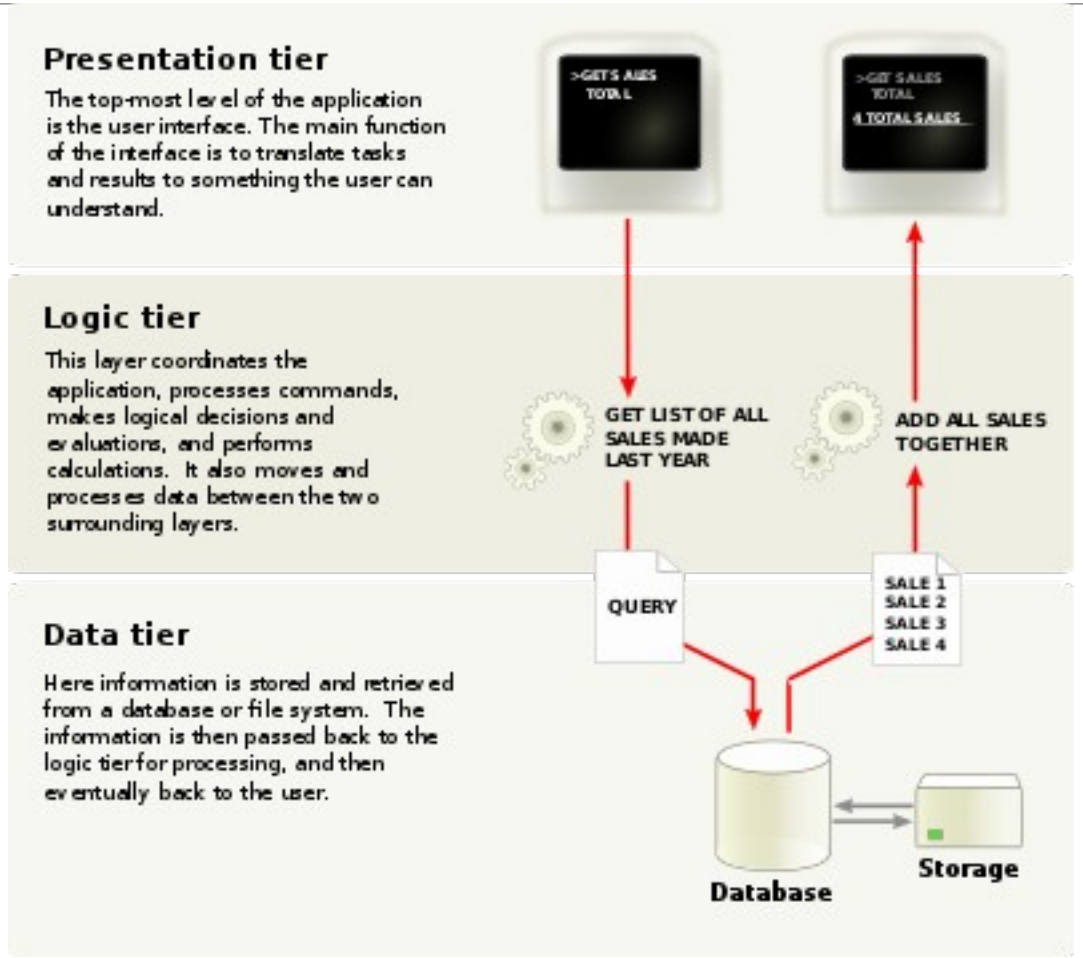
Мікросервісна архітектура

Монолітна

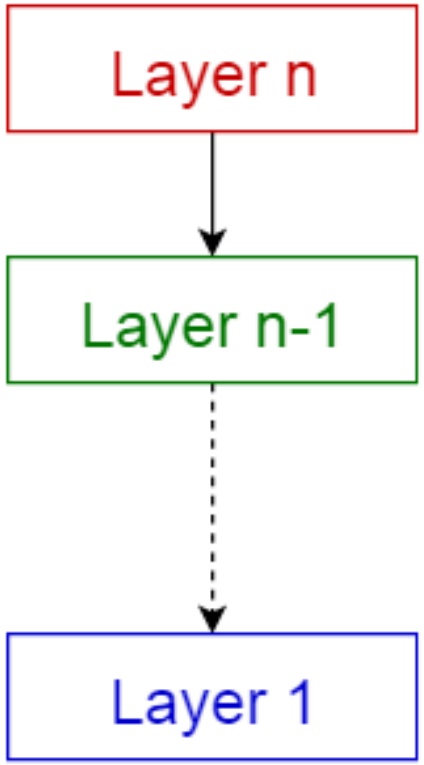
Клієнт-сервер



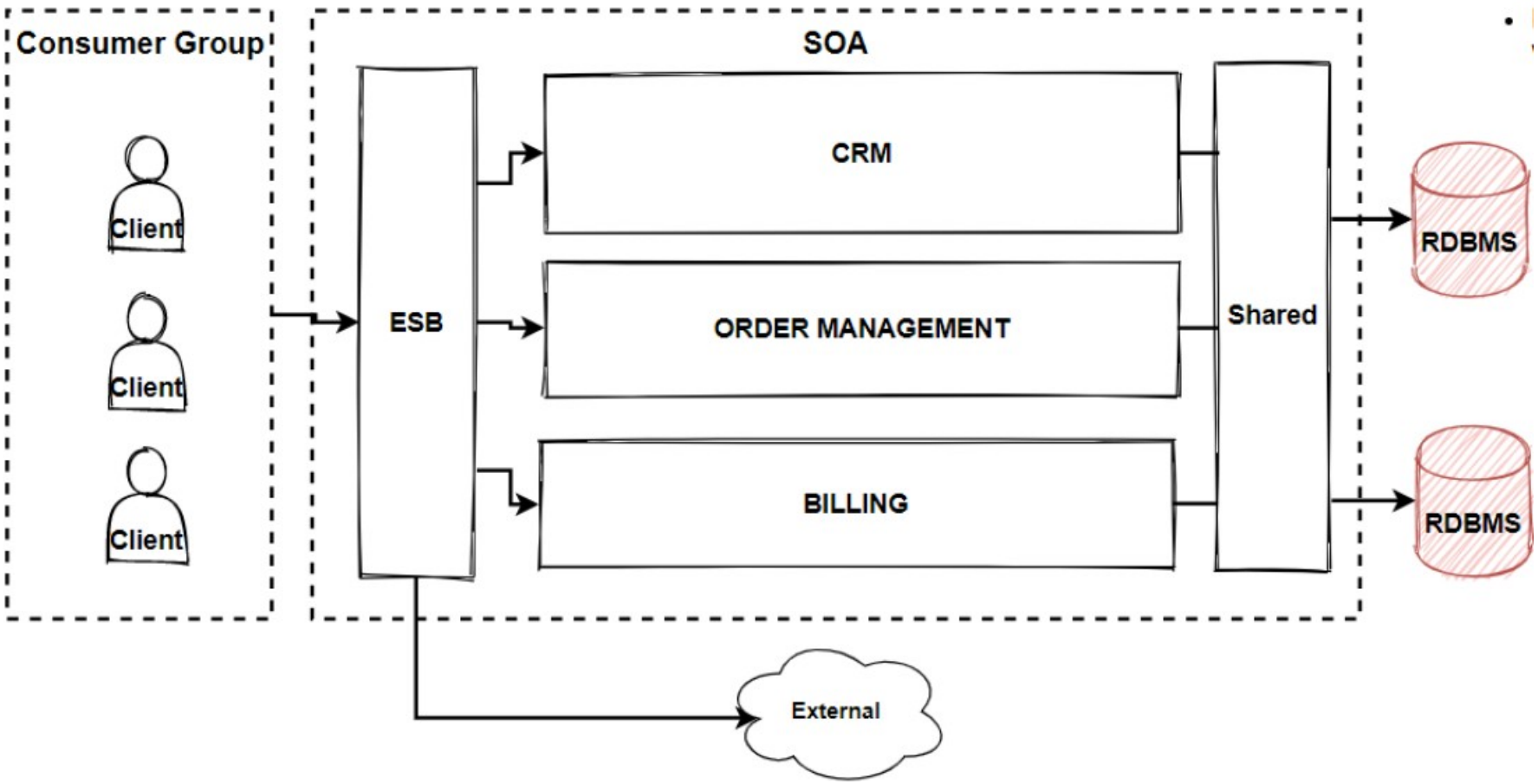
Тришарова


Presentation tier
The top-most level of the application is the user interface. The main function of the interface is to translate tasks and results to something the user can understand.
>GETS ALES TOTAL
>GET SALES TOTAL
4 TOTAL SALES
Logic tier
This layer coordinates the application, processes commands, makes logical decisions and evaluations, and performs calculations. It also moves and processes data between the two surrounding layers.
GET LIST OF ALL SALES MADE LAST YEAR
ADD ALL SALES TOGETHER
Data tier
Here information is stored and retrieved from a database or file system. The information is then passed back to the logic tier for processing, and then eventually back to the user.
QUERY
SALE 1
SALE 2
SALE 3
SALE 4
Database
Storage

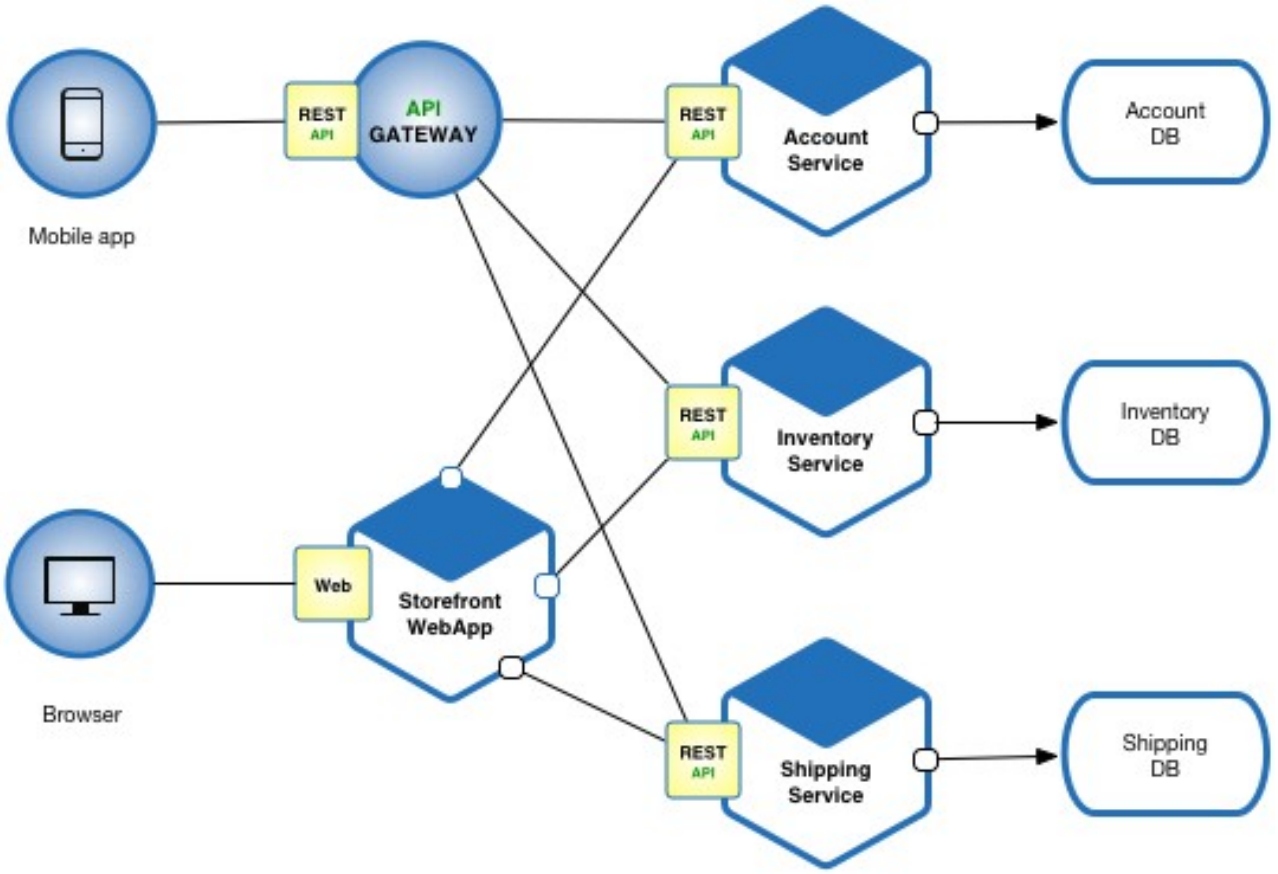
Багатошарова розподілена архітектура



Сервісно-орієнтована архітектура



Мікросервісна архітектура



Стратегії та методи проектування

покрокове уточнення

проектування "зверху-вниз" і "знизу-вгору"

абстракція даних і інкапсуляція

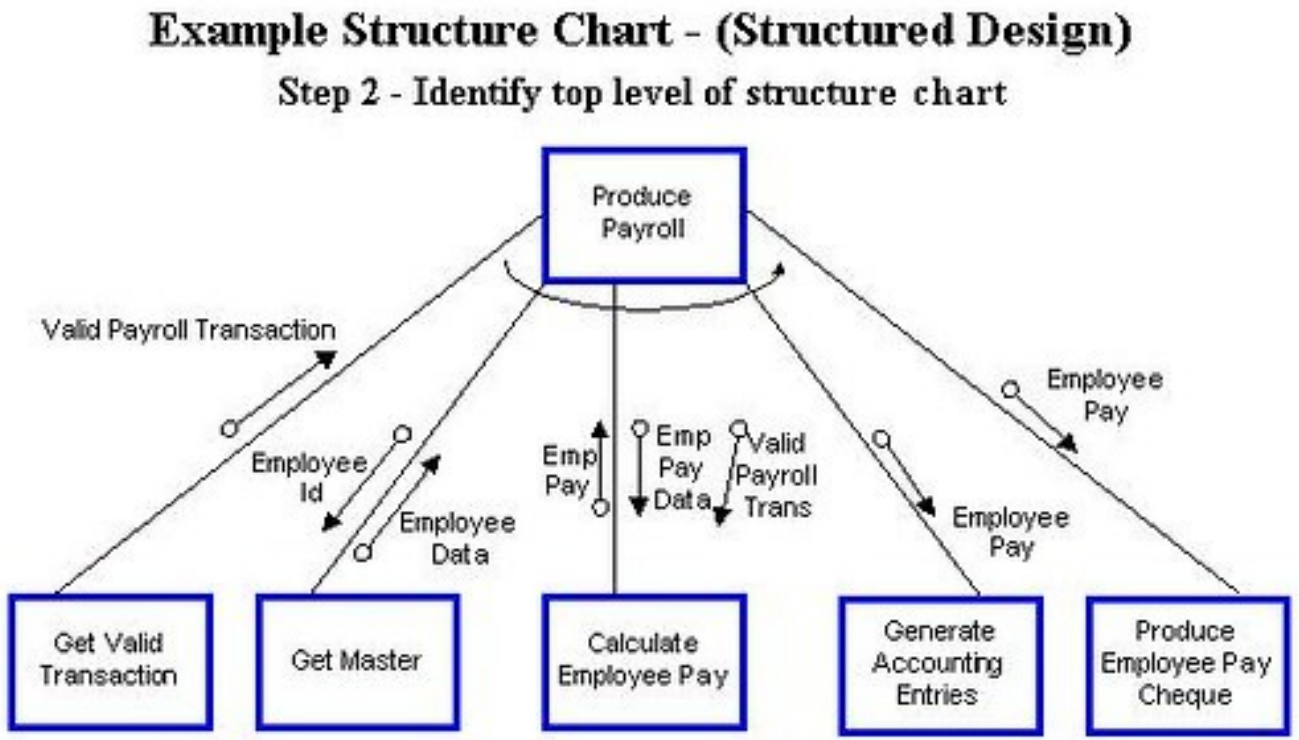
ітеративний підхід

Функціонально-орієнтоване або структурне проектування

Структурне проектування один з класичних методів проектування, в якому декомпозиція сфокусована на ідентифікації основних програмних функцій та детальної розробки та уточнення цих функцій "зверху-вниз«

Структурне проектування використовується після проведення структурного аналізу із застосуванням діаграм потоків даних і описом процесів

Функціонально-орієнтоване або структурне проектування.2

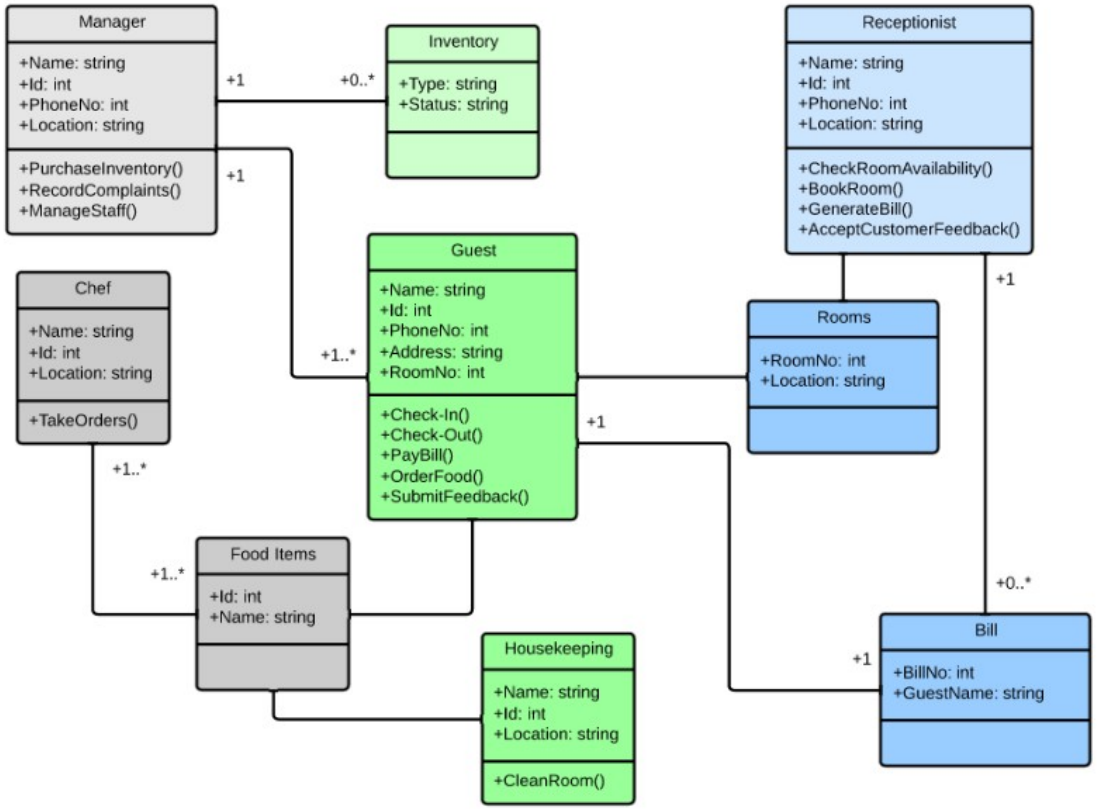


Об'єктно-орієнтоване проектування

Об'єктно-орієнтоване проектування являє собою множину методів проектування, які базуються на концепції об'єктів та класів

У проектуванні часто використовують мову UML

Об'єктно-орієнтоване проектування.2



Розробка (конструювання, Construction)

Основи конструювання

Мінімізація складності

Очікування змін

Конструювання з можливістю перевірки

Стандарти в конструюванні

Мінімізація складності

Мінімізація складності (KISS – Keep It Simple, Stupid) - досягається шляхом надання особливої уваги створенню коду, який є простим та легко читається

Очікування змін

Програмні системи є частиною середовища, що змінюється, і повинні змінюватися разом з ним, а іноді і бути джерелом змін самого середовища

Конструювання з можливістю перевірки

"Конструювання для перевірки" припускає, що побудова програмних систем повинна вестися таким чином, щоб сама програмна система допомагала вести пошук причин збоїв, будучи прозорою для застосування різних методів перевірки, як на стадії незалежного тестування, так і в процесі операційної діяльності - експлуатації

Серед технік, спрямованих на досягнення такого результату конструювання:

• огляд, оцінка коду (code review)

• модульне тестування (unit-testing)

• структурування коду для застосування автоматизованих засобів тестування (automated testing)

• обмежене застосування складних або важких для розуміння мовних структур

Стандарти в конструюванні

Стандарти, які безпосередньо застосовуються при конструюванні, включають:

• комунікаційні методи (наприклад, стандарти форматів документів та оформлення змісту)

• мови програмування і відповідні стилі кодування (наприклад, Java Language Specification, що є частиною стандартної документації JDK - Java Development Kit і Java Style Guide, що пропонує загальний стиль кодування для мови програмування Java, PEP-8 тощо)

• платформи (наприклад, стандарти програмних інтерфейсів для викликів функцій операційного середовища, такі як прикладні програмні інтерфейси платформи Windows - Win32 API, Application Programming Interface або. NET Framework SDK, Software Development Kit)

• інструменти (не в термінах середовищ розробки, але можливих засобів конструювання - наприклад, UML як один із стандартів для визначення нотацій для діаграм, що представляють структуру коду і його елементів або деяких аспектів поведінки коду)

Зовнішні та внутрішні стандарти

Принципи SOLID

Single Responsibility (визначена відповідальність)

Open-Closed (відкриті для розширення та закриті для змін специфікації)

Liskov Substitute (вільне використання підтипів (підкласів) замість суперкласів)

Interface Segregation (багато спеціалізованих інтерфейсів кращі за 1 «суперінтерфейс»)

Dependency Inversion (специфікація залежить від абстракції, а не навпаки)

Практичні міркування

Конструювання - діяльність, в рамках якої програмне забезпечення приводиться до угоди з довільними обмеженнями реального світу

Наближаючись до обмежень реального світу, конструювання (більшою мірою, ніж будь-яка інша область знань) ведеться на основі практичних міркувань і технік:

- Проектування в конструюванні
- Мови конструювання

Проектування в конструюванні

Завжди на стадії конструювання доводиться займатися і питаннями детального дизайну системи

Такі проектні роботи потрібні для слідування стійким обмеженням, нав'язуваним конкретними проблемами, вирішення яких має бути забезпечене використанням програмної системи, що конструюється

Мови конструювання

Мови конструювання включають всі форми комунікацій, за допомогою яких людина може задати вирішення проблеми, що виконується на комп'ютері

Найпростіший тип мов конструювання – конфігураційна мова (configuration language), що дозволяє задавати параметри виконання програмної системи

Інструментальна мова (toolkit language) - мова конструювання з повторно-використовуваних елементів; зазвичай будується як сценарна мова, сценарій (script), виконуваний у відповідному середовищі

Мова програмування (programming language) - найбільш гнучкий тип мов конструювання.

- Містить мінімальний обсяг інформації про конкретні області застосувань та процес розробки, вимагаючи найбільше (у порівнянні з іншими типами мов конструювання) зусиль на вивчення і напрацювання досвіду для ефективного застосування при вирішенні конкретних завдань

Мови конструювання.2

Існує три основних види нотацій використовуваних при визначенні мов програмування:

- лінгвістична
- формальна
- візуальна

Лінгвістичні нотації

Лінгвістичні нотації характеризуються використанням рядків тексту, що містять спеціалізовані "слова", що представляють складні програмні конструкції, і комбіновані в шаблони, що нагадують пропозиції, побудовані у відповідності з певним синтаксисом

У разі коректного використання таких нотацій, кожний одержуваний рядок має строге смислове навантаження (семантику), що забезпечує інтуїтивне розуміння того, що буде відбуватися коли буде виконуватися програмне забезпечення, побудоване з використанням такої мови конструювання

Формальні нотації

Формальні нотації є менш інтуїтивними, ніж лінгвістичні, і часто базуються на точних формальних (математичних) визначеннях

Формальні нотації конструкцій і формальні методи є ядром практично всіх форм системного програмування, точніше - поведінки систем в часі

Такі нотації забезпечують найбільшу готовність одержуваного коду до тестування, що набагато важливіше, ніж просто можливість відображення на звичайній людській мові

Формальні конструкції також використовують точний метод визначення комбінацій застосовуваних символів, що дозволяє уникнути двозначностей, притаманних конструкціям природних мов

Приклад – «Z-нотація»

Візуальні нотації

Візуальні нотації найменш пов'язані з текстово-орієнтованими підходами, припускаючи безпосередню інтерпретацію візуальних конструкцій в процесі виконання описуваної логіки

При цьому логіка в візуальних нотаціях задається розташуванням відповідних візуальних сутностей, відповідальних за ті чи інші операції і структури даних

Приклад - Scratch