

Návrh

architektonický (rozdelenie na moduly) vs. detailný (algoritmy, dátové štruktúry)

(Je to tvorivá činnosť) Vyžaduje netriviálny rozumový vklad. Vymyslieť, ako to čo najlepšie urobiť. Analýza oproti tomu – tiež tvorivá, ale skôr v zmysle čo najlepšie zachytiť niečo, čo tu existuje.

Nebýva robený top-down Návrhár má už isté skúsenosti, isté časti systému vie navrhnuť okamžite, postupuje viac-menej mozaikovito.

Funkčný vs. OO návrh Funkčný návrh: mám funkcie a k nim dáta (centrálne, lokálne).
OO: mám dáta a k nim funkcie.

Modulárnosť

Príklad s CPU

O čo sa snažíme pri návrhu (akomkoľvek):

1. nízka zviazanosť –
 - a. málo rozhraní
 - b. malé rozhrania
 - c. explicitné rozhrania--- spôsob komunikácie: čisto dátami ---

Stupne zviazanosti:

 - a. obsahová
 - b. globálna
 - c. riadiaca
 - d. parciálna dátová
 - e. dátová
2. vysoká kohézia – jeden modul predstavuje:
 - a. jednu funkciu
 - b. jednu, jasne určenú (dátovú) abstrakciuKontrapríklad k (b): NeXT Library mal text class, ktorý zároveň poskytoval možnosti na vizuálne editovanie textu. Používatelia sa sťažovali na to, že táto trieda je príliš veľká – koreň problému však bol v tom, že spájala viacero abstrakcií (reťazec znakov, interaktívne editovateľný text). Rozdeliť to: NSAttributedString a iné, napr. NSTextView.
--- nižšie stupne (k bodu a) ---
 - a. koincidenčná
 - b. logická
 - c. časová
 - d. procedurálna / komunikačná

Ide totiž o:

- modulárnu zrozumiteľnosť (základ pre pochopenie systému, overovanie správnosti, debugovanie),
- modulárnu spojitosť (pružnosť),
- modulárnu ochranu (robustnosť),
- modulárnu komponovateľnosť.