

object oriented analysis

1

attività di analisi

- l'obiettivo dell'analisi è
 - raggiungere la piena comprensione del dominio di interesse
- lo strumento è
 - la descrizione di un modello “di dominio” mediante un opportuno linguaggio
 - il modello deve evidenziare gli aspetti rilevanti e ignorare tutti gli altri

2

modello di dominio

- lo esprimiamo in notazione UML (“domain” class diagram)
- mostra
 - un insieme di classi che rappresentano concetti del dominio
 - associazioni

le classi non rappresentano
classi software!

- i modelli di dominio sono utili anche quando non si deve scrivere software
 - es. business process reengineering

3

altri strumenti

- documenti use case
 - aiutano in una fase iniziale a capire quali sono i concetti fondamentali del dominio
 - svolgono un po’ il ruolo delle specifiche quando il processo non le prevede
 - non le prevede perché non sarebbero affidabili!
 - sequence diagrams utente-sistema
 - coadiuvano gli use case per mostrare gli scenari principali
- glossario
 - affianca il domain class diagram per chiarire ciò che le classi e le loro relazioni rappresentano
 - non è necessario che sia completo, ma è essenziale individuare dove il modello non è chiaro

4

domain class diagram: linee guida

- in ordine di importanza
 - classi
 - associazioni
 - con nome
 - cardinalità
 - stati temporanei inconsistenti trattati in fase di progetto
 - attributi
 - pochi, trasformabili in classi
- alcuni dettagli di UML non hanno senso perché relativi al design
 - operazioni
 - tipi attributi
 - navigabilità
 - visibilità
- alcuni dei dettagli sono poco importanti
 - aggregazione

5

domain class diagram: minimalità

- rappresentare solo ciò che serve al momento con un numero minimo di dettagli
- non esiste il modello perfetto
 - non si deve spendere troppo tempo sul modello
 - il modello verrà raffinato se riconosciuto insufficiente

6

classi: linee guida

- metodo dei nomi
 - i sostantivi degli use case sono candidati per diventare classi
- metodo della checklist
 - oggetti fisici o tangibili
 - descrizioni di oggetti (es. descrizioni di articoli venduti)
 - luoghi
 - transazioni (es. vendita)
 - dettagli o parti di transazioni (es. linee di un ordine)
 - ruoli di persone
 - contenitori (es. scatole di biscotti)
 - cose contenute (es. biscotti)
 - oggetti astratti (es. malattie)
 - organizzazioni (es. società)
 - eventi (es. vendita)
 - processi (es. prenotazioni in corso)
 - regole e politiche (es. sconti)
 - cataloghi
 - registrazioni contabili, legali, ecc. (es. partita doppia)
 - strumenti finanziari (es. obbligazione, azione, linea di credito)
 - documenti (es. manuali, libri, regolamenti)

7

associazioni: linee guida

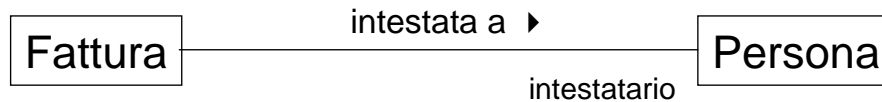
- metodo della checklist
 - A è una parte fisica o logica di B (es. Ala-Aereo,)
 - A è fisicamente o logicamente contenuto in B (es. Libro-Libreria)
 - A è una descrizione di B (es. Articolo-DescrizioneArticolo)
 - A è una linea di una transazione o di un rapporto B (es. LineaFattura-Fattura)
 - A è conosciuto, registrato o riportato in B (Cassa-Vendita)
 - A è un membro di B (es. Pilota-CompagniaAerea)
 - A è una sottounità organizzativa di B (es. Dipartimento-Univesrità)
 - A usa o gestisce B (es. Cassiere-Cassa)
 - A comunica con B (es. Cliente-Cassiere)
 - A è in relazione con una transazione B (es. Cliente-Pagamento)
 - A è vicino a B (es. Città-Città)
 - A è posseduto da B (es. Libro-Libreria)
 - A è un evento relativo a B (es. Partenza-Volo)

8

associazioni: minimalità

- need to know: associazioni che rappresentano informazioni nella realtà da modellare

– es.



- evitare associazioni ridondanti
 - alcune associazioni sono derivabili da altre
- comprehension
 - alcune associazioni sono utili per la comprensione del modello
 - possono anche essere ridondanti
 - trovare un compromesso!

9

attributi: linee guida

- semplici
 - numeri, stringhe, booleani, data, ora, numero di telefono, codice postale, colore, tipi enumerati, codici alfanumerici, ecc.
- gli attributi che rappresentano associazioni sono un problema implementativo
 - in databasisti le chiamano “foreign keys”
- attributi derivati solo se necessari per la comprensione
 - vedi minimalità

10

analisi: esercizio 1

analisi per sistema di prenotazione esami

- Il sistema permette agli studenti di loggarsi, di scegliere l'esame, il canale/professore, di prenotarsi, di annullare una prenotazione.
- Il sistema tiene traccia del periodo in cui la prenotazione è aperta e la data di chiusura.

Dare un modello di dominio per mezzo di un class diagram seguendo le linee guida.

11

analisi: esercizio 2

analisi per sistema di acquisto biglietti cinema

- Il sistema permette ai clienti di acquistare, mediante il numero di carta di credito, biglietti per uno spettacolo in una certa sala
- Ciascuna sala ha una platea con una disposizione differente dei posti. Tale disposizione è conosciuta dal sistema e viene mostrata al cliente all'atto della prenotazione. Ciascun posto è numerato e può ovviamente essere occupato o libero
- Il sistema permette ai clienti di acquistare gruppi di biglietti che devono essere relativi a posti vicini tra loro.
- Un gruppo di biglietti viene ritirato presso una stessa cassa del box office che viene comunicata all'atto dell'acquisto.

Dare un modello di dominio per mezzo di un class diagram seguendo le linee guida.

12