Elemzési, tervezési minták (Analysis/Design Patterns)

Benedek Zoltán (BME-AAIT) fóliáinak felhasználásával

Történelem, személyiségek, hivatkozások

- 1987: OOPSLA'87, Orlando, Ward Cunningham, Kent Beck: „Using pattern languages for Object Oriented Programs" (Smalltalk, GUI)
- 1991: Jim Coplien: Advanced C++ programming styles and idioms.
- Gang of four (G4, GoF, Négyek bandája), Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides különböző könyvek külön majd együtt

Irodalom

- Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides: Design Patterns
  Elements of Reusable Object Oriented Software
  Addison-Wesley 1994.
- James W. Cooper: The Design Patterns Java Companion
  Addison-Wesley Design Pattern Series 1998.
- www.martinfowler.com

Világháló

•Minták Honlapja: *http://hillside.net/patterns/patterns.html*

•A Portland Mintatár: *http://www.c2.com/ppr/*

•Ward Cunningham WikiWiki-je: *http://c2.com/cgi/wiki?WelcomeVisitors*

•Patletek adatbázisa: *http://hillside.net/patterns/patlet/?FrontPage*

•Cetus Links Minták százai: *http://www.objenv.com/cetus/oo_patterns.htm*

•Brad Appleton Szoftver Mintagyűjtemény
http://www.bradapp.net/links/sw-pats.html

•AG Communications Rendszerminták *http://www.agcs.com/patterns/*

•Szervezeti minták honlapja: *http://www.bell-labs.com/cgi-user/OrgPatterns/OrgPatterns*

A (Tervezési) Minták és a speciális relativitáselmélet (avagy „sok igazság van a filozófiában”)

- NEM SZENTÍRÁS!!!
  NEM A BÖLCSEK KÖVE!!!
- NEM CÉL!!! HANEM ESZKÖZ!!!
- MINDEN RELATÍV!!! (ne keverjük össze az ágyút a verébbel vagy a túzokkal!)
- KONKRÉT VISZONYOKHOZ ILLESZTENI KELL!!!
- LÉTEZIK MÁSFÉLE FILOZÓFIA IS!!!

Minták fogalmai

- „Utasításjellegű információrög, ami bizonyos környezetben és erőviszonyok között felmerülő és visszatérő feladatok sikeres megoldáscsaládjainak lényegét és alapelképzelését rögzíti.”
- 3oldalú reláció: „környezet”, „erőviszonyok”, „szoftver konfiguráció”: amely lehetővé teszi, hogy az erők magukat megoldják
- 3oldalú reláció: „környezet”, „feladat”, „megoldás”: a környezet és a feladat által megadott kézenfekvő megoldás
- Egy dolog, ami előfordul – egy utasítás a dolog létrehozására. Egyszerre dolog és folyamat – egy élő dolog és az őt létrehozó folyamat leírása.

Minták fogalma

- A folyamat, ami létrehozza a dolgot, és a dolog, amit a folyamat létrehozott
- 3-as szabály. Legalább 3 környezetben tesztelve…
- Leendő minták: „proto-patterns”, „patlets”
- Anti-patterns: 1. Elrettentő példák 2. Elrettentő helyzetet megfordító példák…
- Best practices… (esettanulmány). Általános / általánosítható / jellegzetes / újrahasznosítható esetek, de ennél SOKKAL TÖBB
- Hasznos, használható és használt!
- „Kérdéses” műfaj. (Az eredmény mellett a filozófia, a megoldáshoz vezető út is érdekes)…

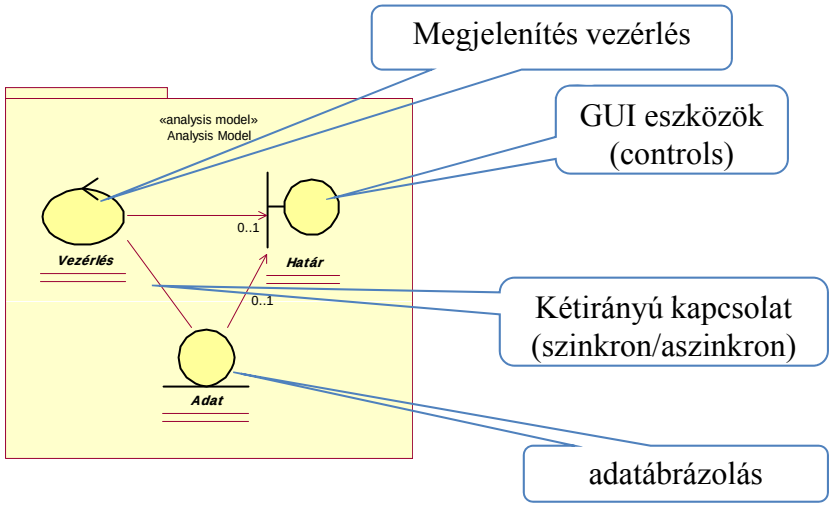
Minták fajtái

- Szervezeti minták – ezzel nem foglalkozunk
- Felépítmény (architectural) minták
- Elemzési minták
- Fogalmi (conceptual) minták
- Tervezési (design) minták
- Kódolási (programkészítési) minták (idiómák)

Tervezési minták fajtái (G4)

- Mi a célja az adott mintának?
  - Létrehozó (creational) minták: nem „egyszerűen” példányosítunk…
  - Szerkezeti (structural) minták: egyszerűbb objektumokból bonyolultabb szerkezetek felépítése
  - Viselkedésmegadó (behavioral) minták
- A minta hatóköre:
  - (Objektum): osztályok kapcsolata, öröklődés
  - (Osztály): objektumok kapcsolata, asszociáció

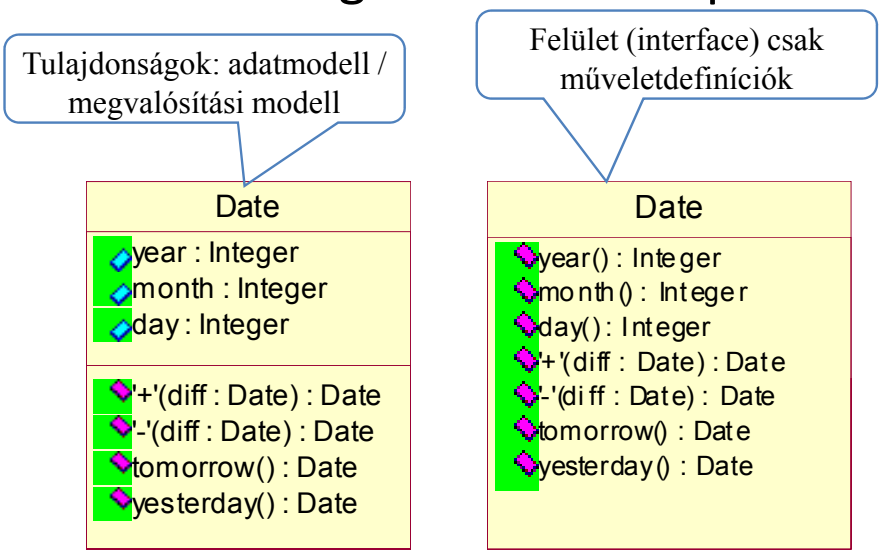
Model-View-Controller (Smalltalk) (Document/View Microsoft)



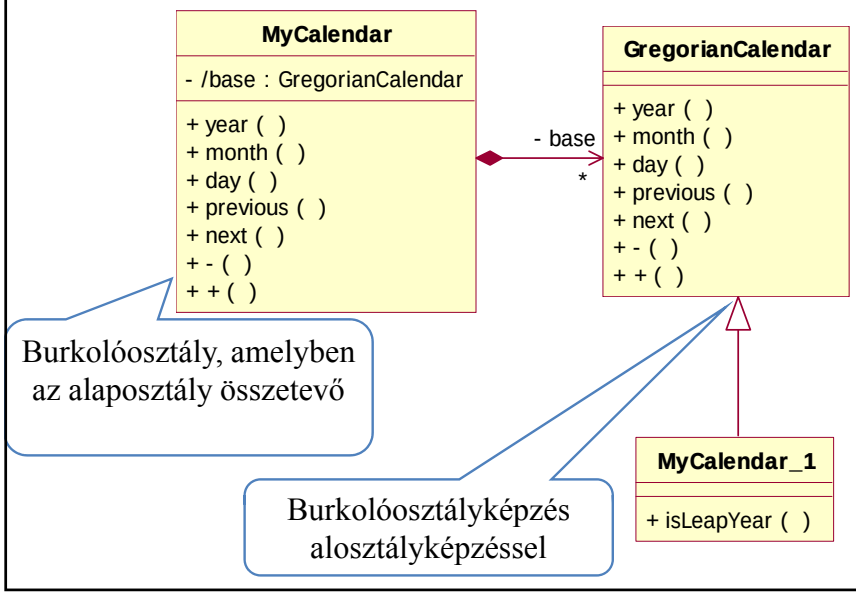
Minta:Időpont

- Időpont ábrázolása valamilyen felbontásban, valamilyen helyen
  - milyen felbontásban? (évjárat ←→ dátum ←→ Timestamp)
  - milyen helyen? (időzónák)
  - műveletek különböző pontosságú adatok között

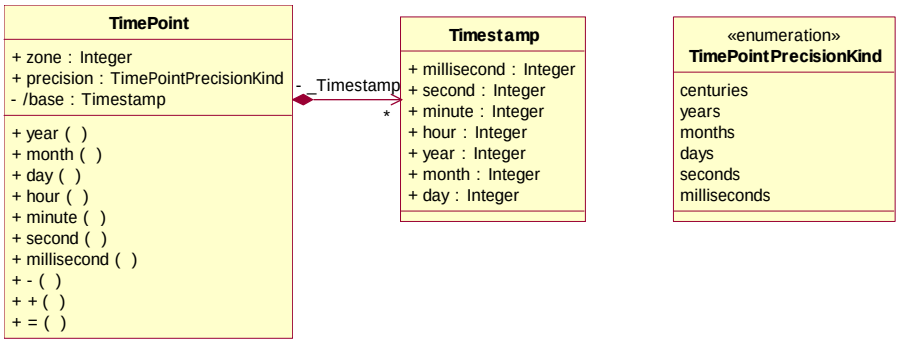
Kétféle megoldás a Date típusra



Burkolóosztályok (adapterek, wrapperek)

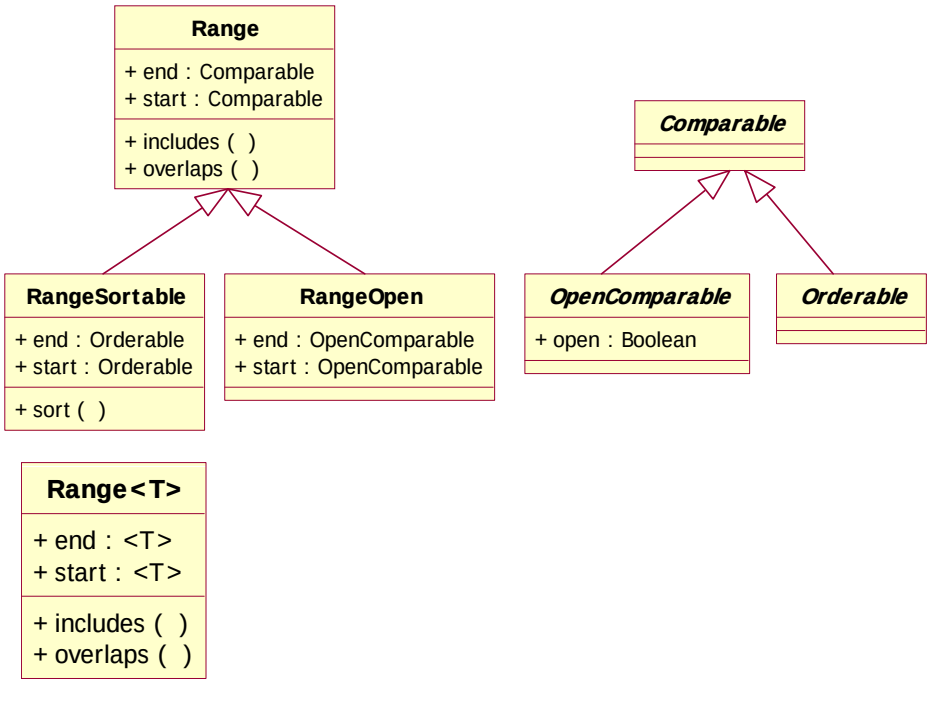


Általános TimePoint specifikációja



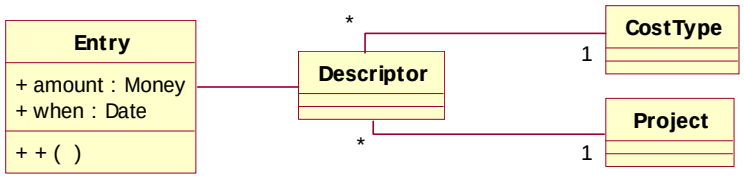
Range (Szakasz)

- Összehasonlítható/rendezhető típusok felett
- Műveletek: szakaszba esés, átlapolás
- Osztálymintákkal (template) megvalósítható
- Sorbarendezés, mint extra művelet
- Nyitott végű szakasz



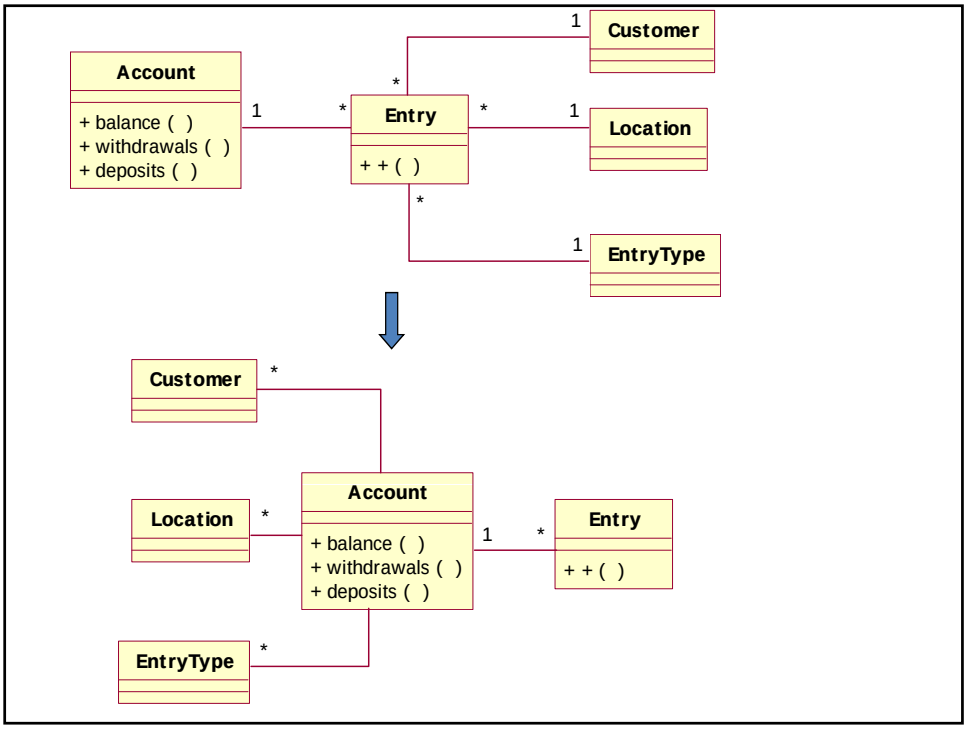
Accounting Entry (Számla tétel)

- Időpont
- Összeg
- Partner
- Helyszín
- Hány számlánk van? (Kinek?)



Account (Számla)

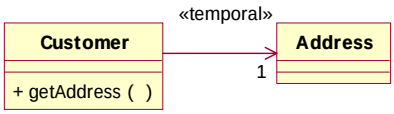
- Tételek halmaza, ahol a tételek összegezhetőek (+(x:Entry):Entry)
- A tételek, a történetük maguk fontosabbak, mint az összegük
- Nem feltétlenül pénztől van szó (fogyasztás, csapadék, teljesítmény stb.)
- Indirekciós lehetőség, szűrhető a tételek többi körülményei alapján



Időfüggés, időgép

- Diszkrét időfüggés: a változás ugrásszerű
- Milyen változásokat kell rögzíteni? Csak jelenidejű változások. Tranzakcióba foglalás
- Csak háttérben tárolt (perzisztens) objektumokra
- Alapesetben a legfrissebb adatokkal dolgozunk
- Az időgép bármilyen korábbi dátumra visszaállítható
- Időbeli változások egyetlen elemre
- Egy konkrét objektumháló (pillanatfelvétel) időben összeegyeztethető (konzisztens)-e
- Időfüggő adatok köre
- Időfüggés finomsága (az egész rendszerre/egy objektumra/egy tulajdonságra)
- Idődimenziók száma (bi-temporal, 2 idődimenzió)

Temporal Property

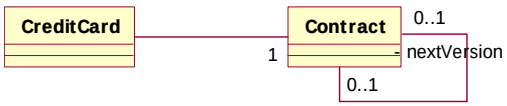


- Lekérdezés (accessor) – időpont függvényében
- Módosítások
  - hozzáfűző – az idősor végén, múltbeli, jelen, jövőbeli
  - módosító – az időszakaszok egymást kizárják?
  - beszúró – egyéb konzisztenciakérdések
  - törlő – (!!!)
- Megvalósítás: 1. egyszerű lineáris vektor 2. Időfüggő gyűjtemény (temporal collection)

Temporal Object

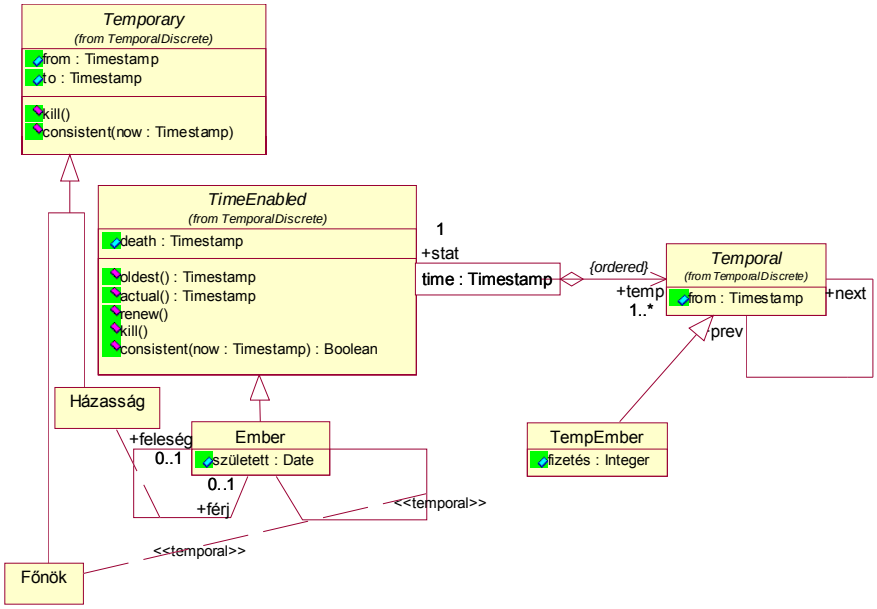
- Időben változó objektum: változás, tartalom ←→ állandóság, identitás.
- Pillanatkép: vagy az egészre, vagy egy tulajdonságra nézve
- Történet, módosítás
- Csak az objektum változik, vagy a hálózat (a kapcsolatok változásait is modellezni kell-e)

Explicit változatok



Időfüggés kétirányú láncolt listával

Egy konkrét megoldás



- Egy időfüggő osztályból 2 osztály (állandó/változó) keletkezik
- TempEnabled alaposztályból származik az állandó, örökli az időfüggés kezelését
- Temporal alaposztályból származik a változó rész
- Egy időfüggő kapcsolat örököl a Temporary kapcsolóosztálytól
- Rendezett, idő szerint indexelt gyűjtemény
- Új változat létrehozása csak aktuális időben, tranzakcióba foglalva/explicit hívásra

Műveletek

- - `oldest`: a művelet előállítja a legöregebb adat időpontját
- - `actual`: a művelet előállítja a legfiatalabb adat időpontját
- - létrehozás (constructor). Az alaposztály létrehozása során az időfüggő vektornak is létrejön a legelső eleme.
- - időfüggő tulajdonságok elérése: az időfüggő objektumra mutató navigációval, az időadat szerinti indexeléssel. Az indexelés algoritmikus megoldása olyan részletkérdés, amelyet nem tárgyalunk.
- - `consistent`: ellenőrzi, hogy a paraméterként megadott időpontban az objektum összes időfüggő összetevője élő-e
- - `renew`: újabb időfüggő példány létrehozása. A példány tulajdonságai a létrehozás után teljesen megegyeznek a korábbi példányéval. A tulajdonságértékek ezek után a tulajdonság-elérő műveletekkel módosíthatók.