Návrhový vzor Composite

Vladimír Jarý
jaryvlad@fjfi.cvut.cz

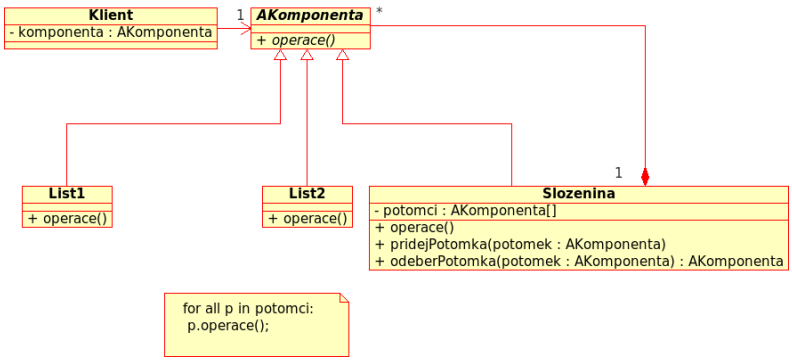
29. října 2007

Abstrakt

Návrhové vzory představují prověřená obecná řešení pro určité třídy problémů spojených s návrhem software. V článku popisovaný vzor *Composite* patří do skupiny strukturálních návrhových vzorů. Tato skupina vzorů popisuje vztahy mezi jednotlivými entitami.

1 Popis vzoru

Návrhový vzor *Composite* umožňuje klientské třídě nebo aplikaci (viz `Klient`) zacházet stejným způsobem s jednoduchými objekty (*komponenty*) a složenými objekty (*kompozita*). Objekty jsou uspořádány do stromové struktury, v kořeni stromu leží abstraktní třída (`AKomponenta`). Tato třída obsahuje definici nějaké abstraktní metody `operace()`. Tato metoda je implementována v komponentách (`List1` a `List2`) a kompozitní třídě (`Slozenina`). Třída složenina si pamatuje množinu komponent ze kterých se skládá (v ukázce se jedná o pole `potomci[]`, ale lze samozřejmě použít i seznam nebo vektor) a obsahuje metody pro přidání a odebrání komponenty. Implementace metody `operace()` spočívá v tom, že se projde seznam obsažených komponent a v každé z nich se zavolá odpovídající implementace.



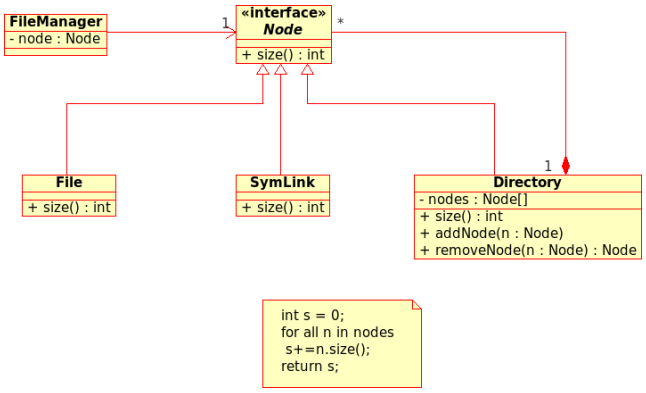
Obrázek 1: UML diagram vzoru

Klient si za běhu udržuje odkaz na instanci zděděné třídy a při volání metody `operace()` se nestará o to, zda se jedná o jednoduchou komponentu či složenou třídu.

Někdy jsou metody pro přidání, respektive odebrání komponenty definovány ve třídě `AKomponenta`. Jejich implementace je potom v komponentách ponechána jako prázdná funkce, případně jako funkce vyvolávající výjimku.

2 Příklad

Návrhový vzor *Composite* se velmi často užívá k reprezentaci rekurzivních datových struktur. Příkladem budiž hierarchický souborový systém (viz obr. 2). Klientem může být správce souborů, který pro jednotlivé elementy souboro-



Obrázek 2: Příklad – souborový systém

vého systému zobrazuje velikost. Konkrétní implementace závisí na tom, zda element je regulární soubor, symbolický odkaz nebo adresář, ale rozhraní `Node` je společné. Regulární soubor a symbolický odkaz představují komponenty, ze kterých se skládá kompozit – adresář. Implementace operace `size()` adresáře pro všechny obsažené komponenty zavolá jejich metodu `size()` a mezivýsledky nasčítá.

Dalším typickým příkladem aplikace vzoru *Composite* je kreslící aplikace. Význam kompozitu má v tomto případě celá kresba skládající se z grafických primitiv (úsečka, kružnice, ...). Operací zde může být nakreslení nebo změna velikosti objektu.

V neposlední řadě se s kompozitem setkáváme při tvorbě okenních aplikací (například knihovna *AWT* nebo *QT*). Jednotlivé komponenty (*widgety*) jako jsou tlačítka nebo vstupní pole se shlukují do složitějších struktur (panely, kontejnery) a ty dále tvoří celé okno.

Reference

[01] Wikipedia, the free encyclopedia
`http://en.wikipedia.org`

[02] M. Dvořák: *Návrhové vzory (design patterns)*
`http://objekty.vse.cz/Objekty/Vzory`

[03] Anti Patterns Catalog
http://c2.com/cgi/wiki?AntiPatternsCatalog

[04] Pattern stories
http://wiki.cs.uiuc.edu/patternStories