



Composite

Kompozit

Skladá objekty do stromových štruktúr reprezentujúcich hierarchie typu časť-celok. Kompozit umožňuje klientom správať sa rovnako k samostatným objektom aj ku kompozíciám.

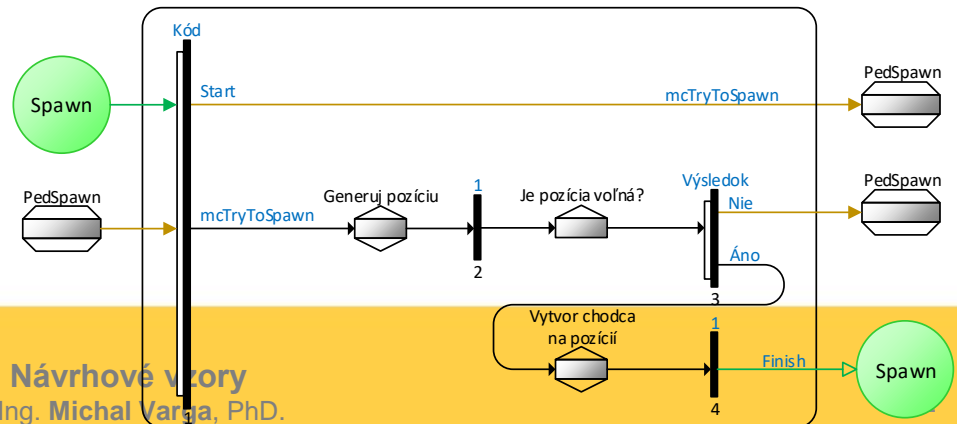
Compose objects into tree structures to represent part-whole hierarchies. Composite lets clients treat individual objects and compositions of objects uniformly.

Návrhové vzory



Motivácia

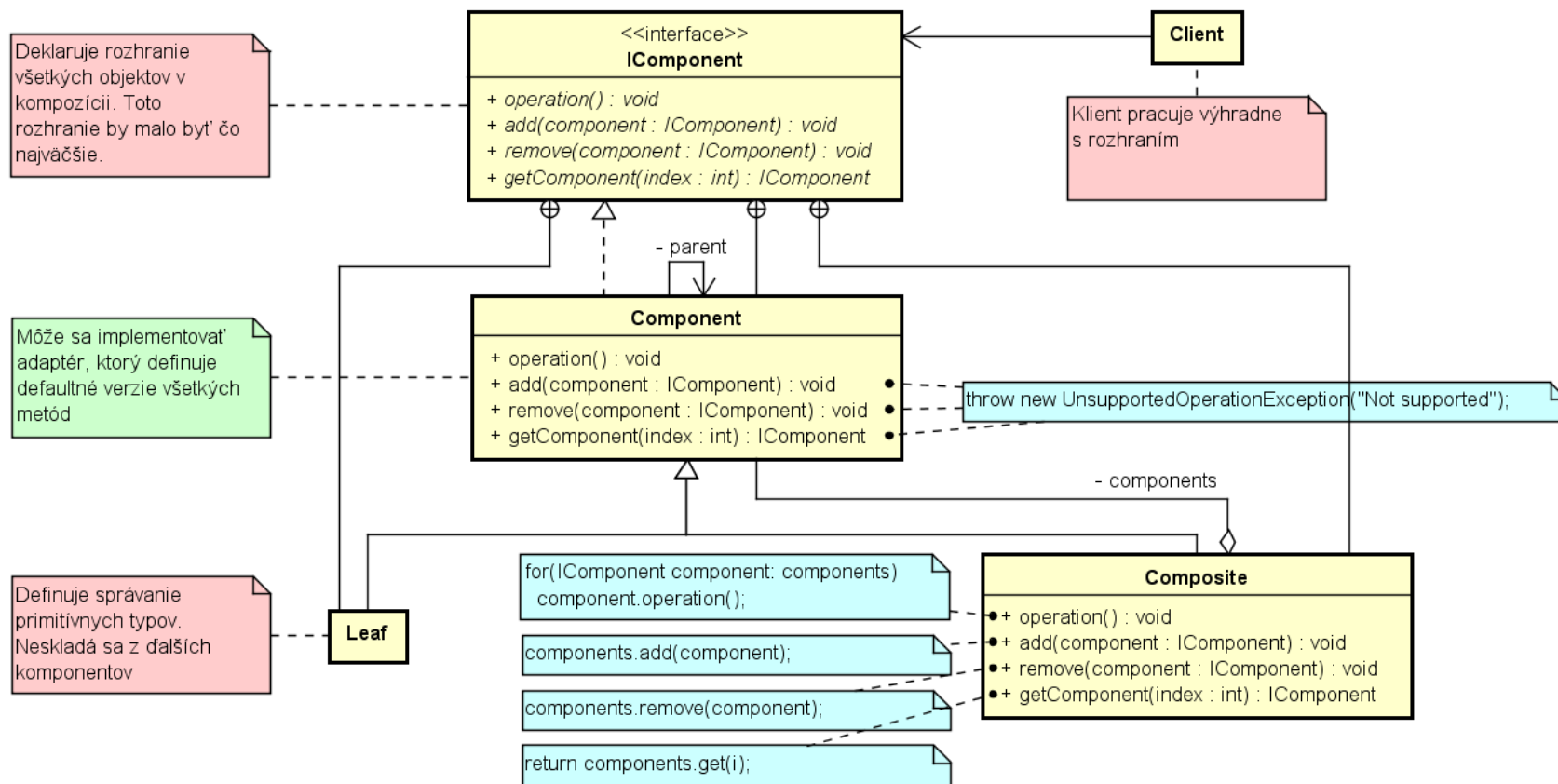
- Majme aplikáciu na kreslenie diagramov. Dokáže kresliť primitívne tvary (čiary, kruhy, obdĺžniky) a tieto následne skladať do rôznych prvkov.
- Každý prvok (či už primitívny alebo zložený) musí byť možné nakresliť, prefarbiť, definovať mu popis, pozíciu, prepojiť ho s iným prvkom atď.
- Ako navrhnuť aplikáciu tak, aby sa ku každému prvku (primitívnemu aj zloženému) správala jednotne?



Aplikovateľnosť

- Kompozit je vhodné použiť, keď:
 - chcete reprezentovať hierarchie typu časť-celok.
 - chcete, aby klienti ignorovali rozdiely medzi celkom a jednotlivcom.

Implementácia



Dôsledky

- Definuje hierarchiu tried pozostávajúcich z primitívnych aj z komplexných typov. Komplexné typy sa skladajú z primitívnych typov, ale môžu obsahovať aj iné komplexné typy. Ak klient očakáva primitívny typ, môže byť ľahko nahradený kompozitom.
- Umožňuje mať jednoduchého klienta, keďže nepotrebuje rozlišovať, či pracuje s primitívnym alebo so zloženým typom.
- Umožňuje jednoducho rozširovať množinu primitívnych aj zložitých typov.
- Zavádza do systému prílišné zovšeobecnenia, čo môže spôsobiť zaťaženie definovania obmedzení na súčasti kompozitu.

Príklad

- Java – Swing, každý prvok je zdedený z `javax.swing.JComponent`, ktorý dedí `Container` = každý prvok môže obsahovať iné prvky.
- Hierarchické údajové štruktúry.

Príbuzné vzory

- *Chain of responsibility* – často sa využíva na vzťahu otec-syn.
- *Decorator* – pri použití s kompozitom často zdieľajú spoločnú rodičovskú triedu, čo spôsobí, že Dekorátor musí implementovať časti rozhrania Kompozitu (add, remove, getChild).
- *Flyweight* – umožňuje zdieľať komponenty, tie však už nemôžu referencovať na rodiča.
- *Iterator* – používa sa na prechody kompozitom.
- *Visitor* – sústreďuje operácie a správanie, ktoré by inak muselo byť distribuované naprieč celým Kompozitom.

Upozornenie

- Tieto študijné materiály sú určené výhradne pre študentov predmetu 5I132 Návrhové vzory (Design Patterns) na Fakulte riadenia a informatiky Žilinskej univerzity v Žiline.
- Reprodukovanie, šírenie (i častí) materiálov bez písomného súhlasu autora nie je dovolené.

Ing. Michal Varga, PhD.
Katedra informatiky
Fakulta riadenia a informatiky
Žilinská univerzita v Žiline
Michal.Varga@fri.uniza.sk

