

TERVEZÉSI MINTÁK

5. forduló



A kategória támogatója: IBM

Ismertető a feladathoz

Felhasznált idő: 00:00/05:00

Elért pontszám: 0/4

1. feladat 0/1 pont

Mely minták elsődleges célja a teljesítménynövekedés elérése?

Válaszok

☒ Object Pool
Ez a válasz helyes, de nem jelölted meg.

☒ Flyweight
Ez a válasz helyes, de nem jelölted meg.

☐ Command

☐ Adapter

☐ Egyik sem

Magyarázat

Az Object Pool erőforrások újrahasznosítását célozza a költséges példányosítások elkerülésével.

A Flyweight az objektumok közös adatainak megosztásával javítja a memóriafelhasználást.

A Command és Adapter mintáknak nem a teljesítménynövekedés az elsődleges célja, de természetesen elképzelhető olyan felhasználási mód, ahol erre is képesek más megoldásokhoz képest.

2. feladat 0/1 pont

Egy logisztikai cég szeretné a termékek dobozba csomagolását egy hierarchikus fastruktúrában modellezni. Minden terméket dobozba tesznek, de a dobozt egy nagyobb dobozba is lehet tenni. Az egymásba dobozolás mélysége tetszőleges. Egy doboz tetszőleges számú elemet tartalmazhat, akár terméket és dobozt is vegyesen.

Ki szeretnének számolni egy doboz értékét is olyan módon, hogy a benne található elemek (és dobozok) értékét összegezzük.

Mely minták alkalmasak erre?

Válasz

☒ Composite
Ez a válasz helyes, de nem jelölted meg.

☐ Binary Tree

☐ Decorator

☐ Builder

☐ Egyik sem

Magyarázat

Ez egy klasszikus példa a Composite képességeire.

A Binary Tree nem tervezési minta, hanem adatstruktúra.

A Decorator célja nem struktúra leképezése, hanem felelőségekkel való dinamikus kiegészítés (továbbá nem is támogatja több elem "csomagolását").

A Builder célja objektumok példányosítása, nem pedig struktúra leképezés.

3. feladat 0/1 pont

Egy objektum műveleteit szeretnénk bővíthetővé tenni úgy, hogy az objektumot ne kelljen ehhez módosítani. Mely minták alkalmasak erre?

Válaszok

☒ Decorator
Ez a válasz helyes, de nem jelölted meg.

☒ Visitor
Ez a válasz helyes, de nem jelölted meg.

☐ Adapter

☐ Composite

☐ Egyik sem

Magyarázat

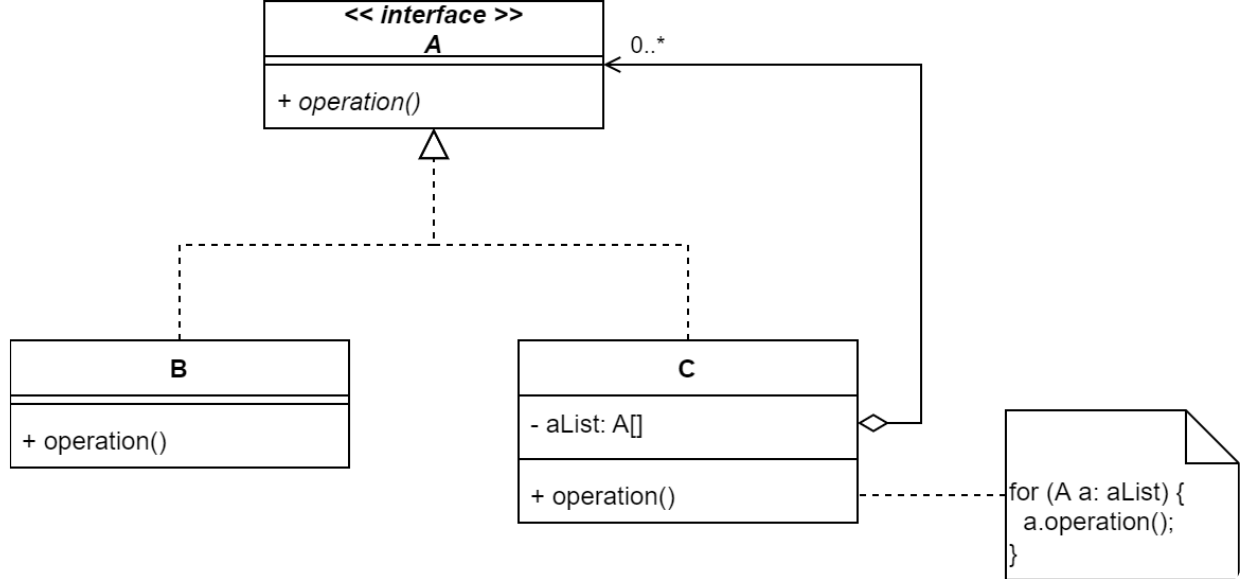
Egy objektum egy (vagy több) Decoratorba csomagolva újabb felelőségekkel ruházható fel.

A Visitor segítségével objektumok egy jól meghatározott csoportjának képességeit tudjuk külső objektumokkal bővíteni.

Az Adapternek és a Composite-nak nem célja új felelősség hozzáadása.

4. feladat 0/1 pont

Az alábbi osztálydiagram melyik mintát reprezentálhatja?



Válasz

☒ Composite
Ez a válasz helyes, de nem jelölted meg.

- ☐ Decorator
- ☐ Proxy
- ☐ Adapter
- ☐ Egyik sem

Magyarázat

A Composite (C) több elem (A) aggregációját támogatja, az operation() végrehajtását pedig delegálja a tartalmazott elemeknek, amik lehetnek egyszerű elemek (B), vagy Composite-ok is.

Az Adapter két inkompatibilis interface megvalósítását célozza, a diagrammon ennek nyoma sincs.

A Decorator és a Proxy esetén egyetlen referenciát tárolunk, nem többet (ráadásul Proxy esetén többnyire nem az absztrakcióra, hanem az implementációra). Továbbá ezen minták lényege, hogy az operation() implementációja a delegálás előtt és/vagy után további műveletekkel van kiegészítve.