

TERVEZÉSI MINTÁK

4. forduló



A kategória támogatója: IBM

Ismertető a feladathoz

A 4. forduló után elérhetőek lesznek a helyezések %-os formában: azaz kiderül, hogy a kategóriában a versenyzők TOP 20% - 40% -60% -ához tartozol-e!

Szeretnénk rá felhívni figyelmedet, hogy a játék nem Forma-1-es verseny! Ha a gyorsaságod miatt kilököd a rendesen haladó versenyzőket, kizárást vonhat maga után!

Felhasznált idő: 00:00/05:00

Elért pontszám: 0/4

1. feladat 0/1 pont

A tervezési minták milyen esetben csökkentik a kód komplexitását?

Válasz

☐ Mindig

☐ Soha

☒ Egyik sem

Ez a válasz helyes, de nem jelölted meg.

Magyarázat

A tervezési minták többnyire új absztrakciós szintet vezetnek be, így nem feltétlenül a legegyszerűbb, hanem a legrugalmasabb megoldást nyújtva. Ebből következően az indokolatlan használatuk fölöslegesen növeli a komplexitást.

De a másik oldal is igaz lehet. Mivel a minták egy konkrét problémára elegáns megoldást adnak, így más megoldásnál egyszerűbb kódot eredményezhetnek.

Így sem a mindig, sem a soha válasz nem lehet igaz.

2. feladat 0/1 pont

Adott egy algoritmus, amit személyre szabhatóvá szeretnénk tenni úgy, hogy bizonyos műveleteket az algoritmus közös részeit tartalmazó osztály leszármazottai valósítanak meg. Mely minták alkalmasak erre?

Válaszok

☒ Factory Method
Ez a válasz helyes, de nem jelölted meg.

☒ Template Method
Ez a válasz helyes, de nem jelölted meg.

☐ Prototype

☐ Strategy

☐ Egyik sem

Magyarázat

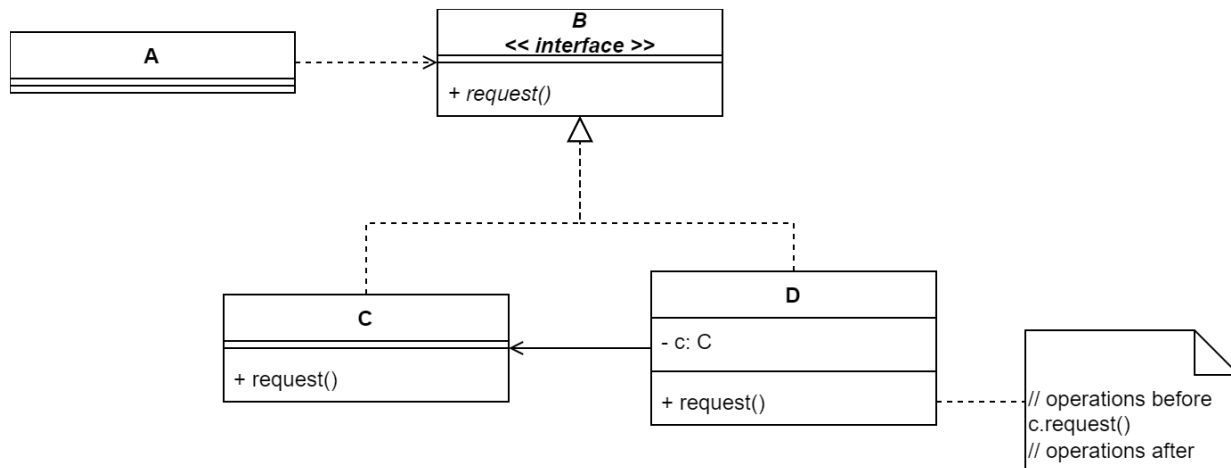
A Factory Method és a Template method esetén egy algoritmus vázát alkotjuk meg, a hiányzó részeket kiszervezzük (többnyire absztrakt) metódusokba, amiket felül tudnak írni a leszármazottak. A különbség, hogy Factory Method esetén az absztrakt metódusok felelőssége valaminek a példányosítása, míg Template Method esetén egyéb jellegű lépések végrehajtása. Így ez a két válasz helyes.

A Strategy-nél nem leszármaztatással szervezzük ki a módosítható részt, hanem kompozícióval, így ez a válasz helytelen.

A Prototype esetén klónozással hozunk létre új objektumokat, így annak teljesen más a célja.

3. feladat 0/1 pont

Az alábbi osztálydiagram melyik mintát reprezentálja?



Válasz

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

☐ Iterator

☐ Bridge

☐ Mediator

☐ Egyik sem

Magyarázat

A Proxy (D) körbeveszi az igazi implementációt (C) oly módon, hogy a végrehajtandó művelet delegálása előtt és/vagy után további műveleteket végezhet. A kliens (A) csak egy absztrakciótól (B) függ, amit az implementáció és a Proxy is megvalósít, így nem látja, hogy a kettő közül melyiket használja.

4. feladat 0/1 pont

Mely állítások igazak a Bridge tervezési mintára?

Válaszok

- ☒ Az absztrakciót és potenciálisan több implementációt választ el egymástól
Ez a válasz helyes, de nem jelölted meg.
- ☐ Potenciálisan több absztrakciót és az implementációt választja el egymástól
- ☐ Az absztrakció dinamikusan cserélhető lehet futási időben
- ☒ Az implementáció dinamikusan cserélhető lehet futási időben
Ez a válasz helyes, de nem jelölted meg.
- ☐ Egyik sem

Magyarázat

Mivel a kliens és az implementációk is az absztrakciótól függenek, így abból egy van és nem cserélhető. Tekintve, hogy az implementáció el van rejtve a kliens elől, akár több is lehet belőle, amik futási időben cserélhetőek is lehetnek.



[Legfontosabb tudnivalók](#) [Kapcsolat](#) [Versenyszabályzat](#) [Adatvédelem](#)

© 2023 Human Priority Kft.

KÉSZÍTETTE **cone**

Megjelenés

Világos