

Tervezési minták egy OO programozási nyelvben beadandó

Pelach Máté Szabolcs | E91K3C

A Kód részletek megtalálhatóak a TMEOOPNY_Kódok.txt fájlban.

Proxy

Egy olyan tervezési minta ami úgy működik hogy egy osztálynak „álcázza” magát, így lehet lekéréseket nézni, előtte, utána.

Pros:

- A használt osztály nem tudja hogy ő egy másik, így könnyen fut le minden át rajta.
- lehet több proxy-t is létrehozni nem csak egyet.

Cons:

- A visszatérés lehet hogy később érkezik

State

A state olyan tervezési minta amely engedélyez egy objektumnak hogy megváltozzon belül, felfogható mintha egy osztályt váltott volna.

Pros:

- Könnyen be lehet újabb státuszt írni anélkül hogy változtatnánk a többin.
- Külön osztályba lehet egy-egy státuszt is tenni.

Cons:

- Értelmetlen ha csak kevés státusza van, vagy esetleg nem változik sokszor

Singleton

A lényege hogy egy olyan osztály amelyet nem lehet instance-olni, de a Singleton-on keresztül lehet használni az osztályait.

Pros:

- Az osztályból csak 1 darab létezhet.
- Minden benne lévő funkció, egyebeket lehet használni

- Akkor lesz az osztály létrehozva hogyha elsőnek használod

Cons:

- Tesztelni Singleton-nal nehéz, vagy esetben nem is lehet.

- MultiThread-be nem hoz létre többet hanem egy marad, érdemes ilyenkor Thread-lockolni
funciókba

MVC (Model View Controller)

Egy olyan tervezési mintát mint a melyet 3 különböző részre osztjuk.

Model, View, Controller.

A kérés a Controllert használva a Model-lel vézi el a műveletet. Majd a Controller egy View-et használva megmutatja mit vázoltott.

Pors:

- Egyidejűleg lehet fejleszteni vele.

- Könnyen változtatható

- Több Model, Viewer is lehet.

Cons:

- Nehezen értelmezhető

- Sok ismétlés

- Sok rész csak az MVC miatt létezik, nem szolgát különösebben sokra.