

Tervezési minták egy OO programozási nyelvben. MVC, - mint modell-nézet-vezérlő minta és néhány másik tervezési minta.

A tervezési minták a programozásban általános megoldások gyakran elforduló problémákra adnak általános újra felhasználható megoldásokat. Problémamegoldáson kívül is hasznosak, ha alkalmazzuk őket kód egyszerűbb és olvashatóbb lesz.

Mivel a minták nem egy programozási nyelvhez vagy specifikus kódhoz kötöttek ezért más nyelvek és technológiákat használóknak is ismerős lehet így könnyebben megértik a kódot még ha nem is ismerik a nyelvet. De ez azt is jelenti, hogy nem tudjuk csak kimásolni és beilleszteni a megoldás.

Általában tartalmazznak egy leírást és egy példát, hogy milyen esetben érdemes használni az adott mintát. Továbbá különböző diagramokat például struktúra diagramot és pszeudokódot és akár kódmintákat a népszerű programozási nyelvekhez tartalmazhatnak.

Modell-nézet-vezérlő:

Ez a minta nagyon hasznos ha szeretnénk a jól elkülöníteni a felhasználói felületet a üzleti logikát és függetlené tenni azt az adatszerkezetől. Kezdetben az asztali alkalmazásoknál volt népszerű. Manapság sok nyelvnek mint a JavaScript, Python, Ruby, PHP, Java, C# és Swift van beépített MNV keretrendszerük. Hasznosága és elérhetősége miatt már web és mobilalkalmazások nagyrészt alkalmazásával fejlesztik

Az MNV több rétegre bontja az alkalmazás, mint a tradicionálisabb módszerek így az egész alkalmazás szerkezetét diktálja jóval meghatározóbb ily szempontból mint a legtöbb tervezési minta.

A rétegek:

Modell: Az adatokat kapcsolatos logikát végzi és kezeli az adatbázist

Nézet: maga a felhasználói felület prezentálja az adatot a felhasználó felé dinamikus, mindig a legfrissebb adattal dolgozik egy modellhez több nézet is tartozhat

Vezérlő: Összeköti a modellt és a nézetet

Singleton (egyke):

Egyetlen példányra korlátozza a létrehozható példányoknak a számát. Ezt el tudjuk érni ha nincs vagy privát a konstruktorunk és a helyett egy olyan metódust hívunk ami ha nincs még példányosítva a class akkor létrehoz egy példányt és vissza adja azt ha már létre lett hozva akkor a helyet hogy újat készítené vissza adja a régit.

gyártó metódus:

Ezzel a mintával le egyszerűsíthetjük az osztályok példányosító utasításait ha sok hasonló objektumot kell létre hoznunk melyeknek sok paraméterük van ezt automatizálhatjuk.

Prototípus:

Segítségével mély klónozást hajthatunk végre és nem csak a hivatkozást másoljuk, hanem az értékeket is