

Tervezési minták egy OO programozási nyelvben.

MVC, mint modell-nézet-vezérlő minta és néhány másik tervezési minta.

Az informatikában rendszerint tervezési mintáknak nevezik azokat a gyakran előforduló programozási feladatokra alkalmazható általános, újrafelhasználható megoldásokat. Egy programtervezési minta általában objektumok és osztályok közötti együttműködést ír le.

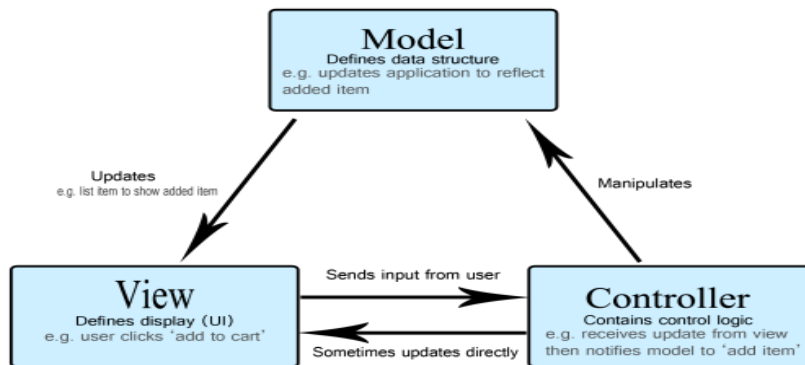
A tervminták nem kész terveket szolgáltatnak, amelyeket közvetlenül le lehetne kódolni. Ugyanakkor példakódokat is tartalmaznak, amelyeket az adott helyzetre alkalmazott kóddal kell kitölteni. Céljuk a leírás vagy sablon nyújtása, amely segíti a megoldás formalizálását.

A minták általában osztályok és objektumok közötti kapcsolatokat mutatnak be, de nem specifikálják konkrétan a végleges osztályokat vagy objektumokat. Egyes esetekben interfészek is alkalmazhatók az absztrakt osztályok helyett, bár ezeket maga a tervminta nem mutatja meg. Néhány programozási nyelv már beépítve tartalmaz tervezési mintákat. A tervminták a strukturált programozás egy szintjeként tekinthetők, amely a paradigma és az algoritmus között helyezkedik el.

- létrehozási minták
- szerkezeti minták
- viselkedési minták

MVC

Az MVC egy összetett tervezési minta, ami a grafikus felülettel rendelkező alkalmazásoknál használható, főleg WEB-es környezetben alkalmazzák. Gyakori egy alkalmazás több rétegre svaló felbontása: **megjelenítés** (felhasználói felület), **tartománylogika** és **adatelérés**. Az MVC-ben a megjelenítés tovább bomlik nézetre és vezérlőre. Az MVC sokkal inkább meghatározza egy alkalmazás szerkezetét, mint az egy programtervezési mintára jellemző.



Modell

Az alkalmazás által kezelt információk tartomány-specifikus ábrázolása. A tartománylogika jelentést ad a puszta adatnak (pl. kiszámolja, hogy a mai nap a felhasználó születésnapja-e, vagy az összeget, adókat és szállítási költségeket egy vásárlói kosár elemeihez).

Sok alkalmazás használ állandó tároló eljárásokat (mint mondjuk egy adatbázis) adatok tárolásához. Az MVC nem említi külön az adatelérési réteget, mert ezt beleérti a modellbe.

Nézet

Megjeleníti a modellt egy megfelelő alakban, mely alkalmas a felhasználói interakcióra, jellemzően egy felhasználói felületi elem képében. Különböző célokra különböző nézetek létezhetnek ugyanahhoz a modellhez.

Vezérlő

Az eseményeket, jellemzően felhasználói műveleteket dolgozza fel és válaszol rájuk, illetve a modellben történő változásokat is kiválthat.

Néhány további tervezési minták

1. Létrehozási minták:

Singleton minta:

Egyetlen példány létrehozása biztosítása egy osztályból

Factory minta:

Objektum létrehozás kiszervezése egy másik osztályra.

Builder minta:

Összetett objektumok létrehozásának egyszerűsítése és lépésről lépésre történő konfiguráció

2. Szerkezeti minták:

Decorator minta:

Dinamikusan bővíthető objektumok létrehozása

Adapter minta:

Két inkompatibilis rendszer egyesítése úgy, hogy azok együttműködjenek.

Composite minta:

Egységes interfész kialakítása az egyszeres és összetett objektumok számára.

3. Viselkedési minták:

Observer minta:

Az objektumok közötti egy-egy kapcsolat kialakítása, úgy hogy ha az egyik változik, a másik értesüljön róla.

Strategy minta:

Az algoritmusokat leválasztani a fő osztályról, és azokat cserélhetővé tenni.

Command minta:

Egy kérés objektummá alakítása, hogy paraméterezhetővé, sorban végrehajthatóvá és visszavonhatóvá váljon.