

TERVEZÉSI MINTÁK AZ OBJEKTUMORIENTÁLT PROGRAMOZÁSBAN ÉS AZ MVC BEMUTATÁSA

A tervezési minta (Design Patterns) leír egy gyakran előforduló programtervezési problémát, annak környezetét és a megoldás magját, amit alkalmazva számos gyakorlati eset hatékonyan megoldható. Rendszerint egymással együttműködő objektumok és osztályok leírása.

A minták által a rendszer egyszerűbb, karbantarthatóbb és ami a legfontosabb, hogy többször felhasználható lesz, másfelől a minták ismeretével megérteni is könnyebb a rendszert, ezáltal egy-egy rész megértése a különféle nyelvek kódjaiban könnyebbé válik.

Tervezési minták leírása

Minden tervezési minta leírásához négy alapelem tartozik

- ✓ Pattern név (pattern name)
- ✓ Probléma (problem)
- ✓ Megoldás (solution)
- ✓ Következmények (consequences)

A tervezési minták csoportjai

Létrehozási tervezési minták

A megfelelő objektumokkal és a létrehozás mechanizmusával foglalkoznak

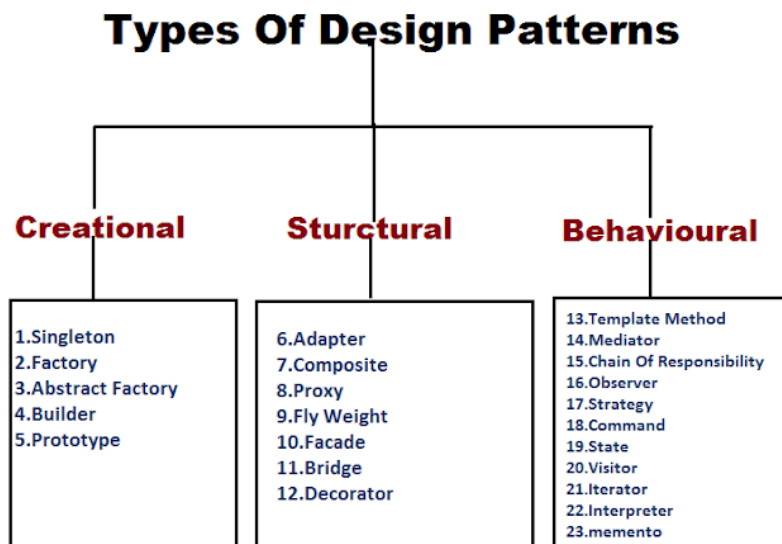
Strukturális tervezési minták

Megkönnyítik a tervezést az entitások közötti kapcsolatok megvalósításának egyszerű módjának azonosításával

Viselkedési tervezési minták

Előtérbe helyezi az osztályok objektumkommunikációját.

<i>Létrehozási</i>	<i>Tervezési</i>	<i>Viselkedési</i>
Factory method	Adapter	Command
Abstract factory	Bridge	Memento
Singleton	Composite	Template
Prototype	Decorator	Strategy



A tervezési minták előnyei

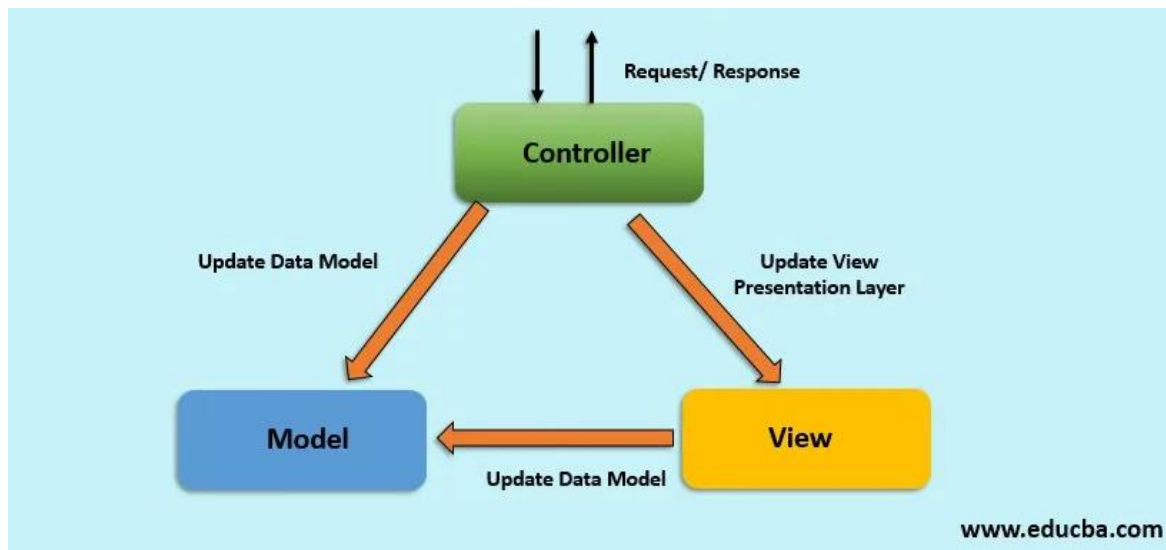
A tervezési minták

- ✓ közös nyelvet használnak
- ✓ megkönnyítik a fejlesztők számára a problémákkal kapcsolatos kommunikációt
- ✓ a tervezés korai szakaszában jobb a kód olvashatósága és architektúrája

MVC

(tervezési minta)

A Model–View–Controller (MVC) egy felhasználói felületek fejlesztésére általánosan használt szoftvertervezési minta, amely a kapcsolódó programlogikát három, egymással összefüggő elemre osztja. Ezek az elemek az információ belső reprezentációi (a modell), az információt megjelenítő és a felhasználótól fogadó felület (nézet), valamint a kettőt összekötő vezérlő szoftver. (Wikipedia)



Az MVC előnyei:

- ✓ Egyidejű fejlesztés – Több fejlesztő tud egyszerre külön a modellen, vezérlőn és a nézeteken dolgozni.
- ✓ Magas szintű összetartás – MVC segítségével az összetartozó funkciók egy vezérlőben csoportosíthatóak. Egy bizonyos modell nézetei is csoportosíthatóak.
- ✓ Függetlenség – MVC mintában az elemek alapvetően nagy részben függetlenek egymástól
- ✓ Könnyen változtatható – Mivel a felelősségek szét vannak választva a jövőbeli fejlesztések könnyebbek lesznek
- ✓ Több nézet egy modellhez – Modelleknek több nézetük is lehet
- ✓ Tesztelhetőség – Mivel a felelősségek tisztán szét vannak választva, a külön elemek könnyebben tesztelhetők egymástól függetlenül

Az MVC jellemzői

- ✓ Könnyű és súrlódásmentes tesztelhetőség.
- ✓ Kiválóan tesztelhető, bővíthető és csatlakoztatható keretrendszer
- ✓ Az MVC-mintát használó webalkalmazás-architektúra megtervezéséhez
- ✓ A pályázati feladatok szétválasztása pl. üzleti logika, UI logika és bemeneti logika
- URL-útválasztás SEO-barát URL-ekhez. Hatékony URL-leképezés az érthető és kereshető URL-ekért

Ismert és széleskörűen MVC keretrendszerek:

- ✓ Ruby on Rails
- ✓ Django
- ✓ Spring MVC
- ✓ Rails
- ✓ Laravel

