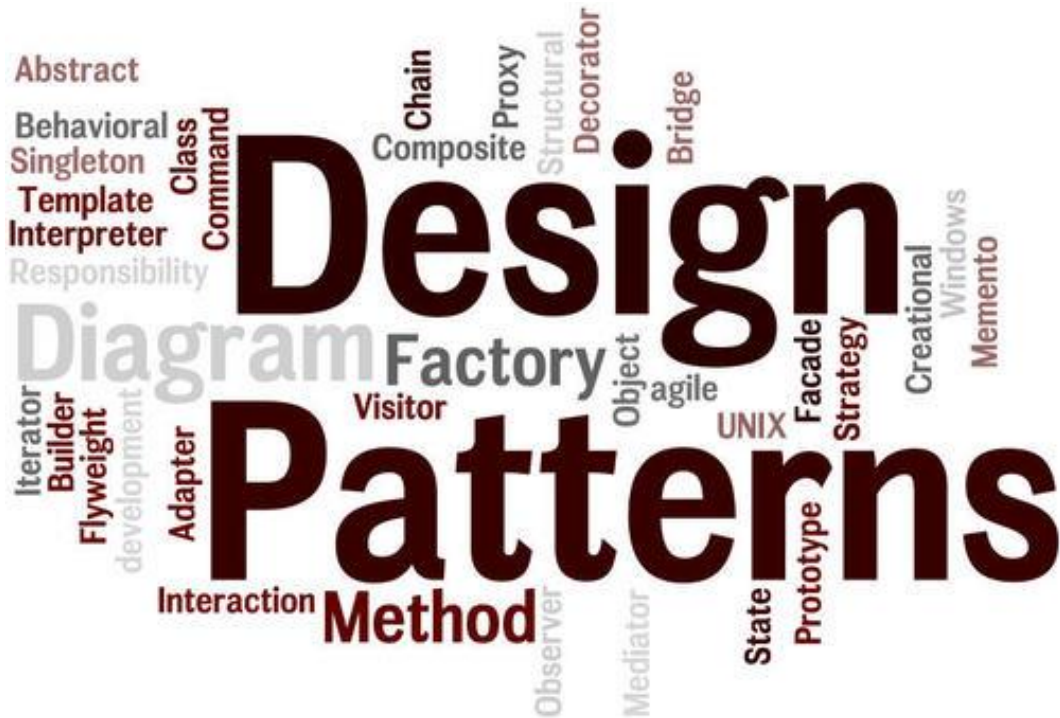


Tervezési minták

Az objektumorientált programok elemeinek újrahasznosítása és a létrehozási minták rövid bemutatása



„Minden minta olyan problémát ír le, ami újra és újra felbukkan a környezetünkben és aztán leírja a megoldás magját, oly módon, hogy a megoldás milliószor felhasználható legyen anélkül, hogy valaha is kétszer ugyanúgy csinálnánk”

Christopher Alexander

Tartalomjegyzék

| | |
|--|---|
| A tervezési minták általános tulajdonságai | 2 |
| Minta neve..... | 2 |
| Probléma | 2 |
| Megoldás | 2 |
| Következmények | 2 |
| Legegyszerűbb és legáltalánosabb tervezési minták | 2 |
| A tervezési minták osztályozása..... | 3 |
| Létrehozási minták | 3 |
| Elvont gyár (Abstract Factory) - objektum létrehozási minta..... | 4 |
| Építő (Builder) – objektum-létrehozási minta..... | 4 |
| Gyártófüggvény (Factory Method) – osztálylétrehozási minta..... | 5 |
| Prototípus (Prototype) – Objektum-létrehozási minta | 5 |
| Egyke (Singleton) – objektum-létrehozási minta..... | 6 |
| Felhasznált irodalom..... | 7 |

A tervezési minták lényege, hogy ne legyen szükség minden problémára új megoldásokat kidolgozni, hanem célszerűbb újrahasznosítani azokat a megoldásokat, amelyek korábban már működőképesnek bizonyultak. A tervezési minták egymással együttműködő objektumok és osztályok leírásai, amik testreszabott formában valamilyen általános tervezési problémát oldanak meg egy bizonyos összefüggésben.

Adott tervezési minták adott tervezési problémákra nyújtanak megoldást és alkalmazásukkal rugalmassá és újrahasznosíthatóvá válnak az objektumorientált rendszereink, alkalmazásaink.

A tervezési minták általános tulajdonságai

Minta neve

Egy hivatkozási eszköz, amit egy tervezési probléma és annak megoldásának, következményeinek leírására használunk. Segít abban, hogy tudjunk gondolkodni a tervezésről, tudjunk beszélni a megoldandó problémáról.

Probléma

Elmagyarázza a problémát és összefüggéseit. Leírhat konkrét tervezési problémákat, de néha feltételek listáját is tartalmazhatja, amiknek teljesülnie kell, mielőtt alkalmaznánk a mintát.

Megoldás

Azokat az elemeket írja le viszonyaikkal, hatáskörökkel és együttműködési lehetőségeikkel együtt, amelyek felépítik a tervet. Nem ad meg konkrét tervet vagy megvalósítást, mivel a minta egy sablon, ami különböző helyzetekben alkalmazható.

Következmények

A minta alkalmazásának előnyei és hátrányai. Létfontosságúak a tervezési alternatívák értékelésénél és annak megértésében, hogy mik előnyei és a hátrányai a minta alkalmazásának. A következmények erőforrás-felhasználást (tárhely, idő, memória) érintő hátrányokra vonatkoznak, de érinthetnek megvalósítási vagy nyelvi kérdéseket is. Az újrahasznosíthatóság miatt a következmények felsorolása segít megérteni és értékelni az előnyöket és hátrányokat.

Leggyakoribb és legáltalánosabb tervezési minták

- Elvont gyár – Abstract Factory
- Illesztő – Adapter
- Összetétel – Composite
- Díszítő – Decorator
- Gyártófüggvény – Factory Method
- Megfigyelő – Observer
- Stratégia – Strategy
- Sablonfüggvény – Template Method

Ezek a leggyakrabban alkalmazott minták segítenek megérteni az objektumorientált tervezés lényegét, az egyes tervezési minták sajátosságait.

A tervezési minták osztályozása

A tervezési minták részletességükben és elvonatkoztatási szintjük szerint különbözőek. A tervezési mintákat hatókörük és céljuk alapján a következő ábrán látható módon rendszerezhetjük.

| | | Cél | | |
|---------|----------|---|---|---|
| | | Létrehozási | Szerkezeti | Viselkedési |
| Hatókör | Osztály | Gyártófüggvény | (Osztály)illesztő | Értelmező Sablonfüggvény |
| | Objektum | Elvont gyár Építő Prototípus Egyke | (Objektum)illesztő Híd Összetétel Díszítő Homlokzat Pehelysúlyú Helyettes | Felelősséglánc Parancs Bejáró Közvetítő Emlékeztető Megfigyelő Állapot Stratégia Látogató |

A tervezési minták osztályozásakor a mintáknak létrehozási, szerkezeti vagy viselkedési célja lehet.

- A létrehozási vagy alkotó minták az objektum-létrehozás folyamatában érdekeltek.
- A szerkezeti minták az objektumok és osztályok felépítésével foglalkoznak.
- A viselkedési minták azokat a tulajdonságokat írják le, ahogyan az objektumok kölcsönösen együttműködnek egymással és felosztják a felelősségi köröket.

A hatókör az, ami meghatározza, hogy a minta objektumra vagy osztályra alkalmazható-e.

- Az osztályminták az osztályok és az alosztályok viszonyával foglalkoznak. Ezek a viszonyok öröklődéssel létesítődnek, így statikusak és fordításkor rögzítettek.
- Az objektumminták objektumkapcsolatokra vonatkoznak, amik futásidőben megváltoztathatók, ezért jóval dinamikusabbak.

Létrehozási minták

A létrehozási minták a példányosítási folyamat elvont ábrázolásai. Függetlenítik a rendszert a benne lévő objektumok létrehozási módjától, összeállításától és megjelenítésétől. Az osztálylétrehozási minták öröklés útján módosítják azt az osztályt, amelyből példány készül, míg az objektum-létrehozási minták átruházzák a példányosítást egy másik objektumra.

Amikor a rendszer elkezd jobban függeni az objektumok összeállításától, mint az osztályörökléstől, akkor a létrehozási minták egyre fontosabbá válnak. Ekkor a lényeg áthelyeződik a rögzített viselkedéshalmaz állandó bekódolásáról az alapvető viselkedésmódok kisebb halmazának meghatározására. Ezeket a kisebb viselkedéshalmazokat később akárhány összetettebb halmazba is beépíthetjük.

Ezekbe a mintákba be van ágyazva, hogy mely osztályokat használja viszont el van rejtve, hogy ezen osztályok példányait hogyan hozza létre és rakja össze a rendszer. Ennek köszönhetően

a létrehozási minták nagy rugalmasságot tesznek lehetővé, hogy mit, mikor, ki és hogyan hoz létre. A létrehozási minták kialakítása lehet statikus, azaz fordítási időben megadott vagy dinamikus, azaz futási időben megadott.

Elvont gyár (Abstract Factory) - objektum létrehozási minta

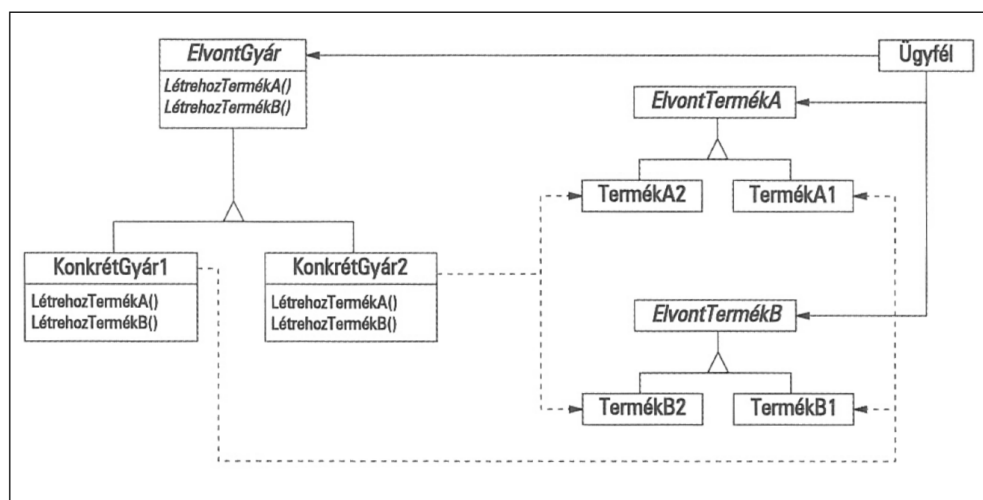
Cél

Kapcsolódó vagy egymástól függő objektumok családjának létrehozására szolgáló felületet biztosítani konkrét osztályok számára.

Alkalmazhatóság

- A rendszernek függetlennek kell lennie attól, hogy a „termékeit” hogyan hozza létre, állítja össze és jeleníti meg.
- A rendszert úgy kell beállítani, hogy a „termékcsaládok” egyikével használható legyen.
- Az egy családhoz tartozó termékobjektumokat együttes használatra tervezték és ezt a megszorítást ki kell kényszerítenünk.
- Termékek osztálykönyvtárát szeretnénk létrehozni, de csak a felületeiket szeretnénk felfedni, megvalósításukat nem.

Szerkezet



Építő (Builder) – objektum-létrehozási minta

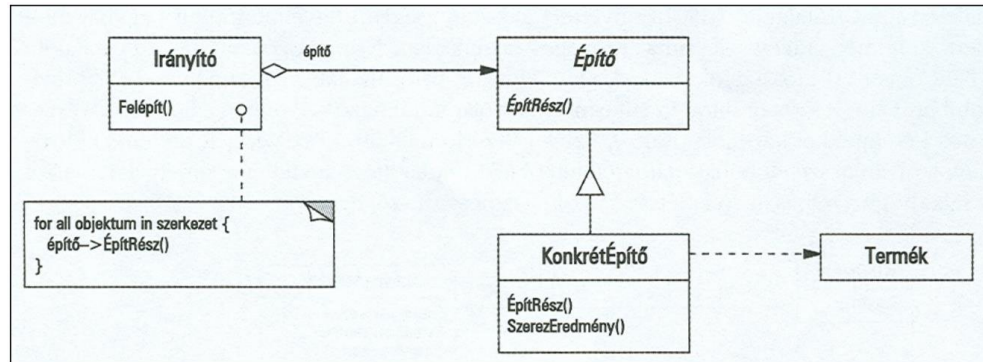
Cél

Az összetett objektumok felépítésének függetlenítése az ábrázolásuktól, így ugyanazzal az építési folyamattal különböző ábrázolásokat hozhatunk létre.

Alkalmazhatóság

- Az összetett objektumok létrehozási algoritmusának függetlennek kell lennie az objektumot alkotó részegységektől és azok összeállítási módjától.
- Az építés folyamatának lehetővé kell tennie, hogy a létrehozott objektum többféleképpen jelenhessen meg.

Szerkezet



Gyártófüggvény (Factory Method) – osztálylétrehozási minta

Cél

Felület meghatározása egy objektum létrehozásához, az alosztályokra bízva, melyik osztályt példányosítják. A gyártófüggvények megengedik az osztályoknak, hogy a példányosítást az alosztályokra ruházzák át.

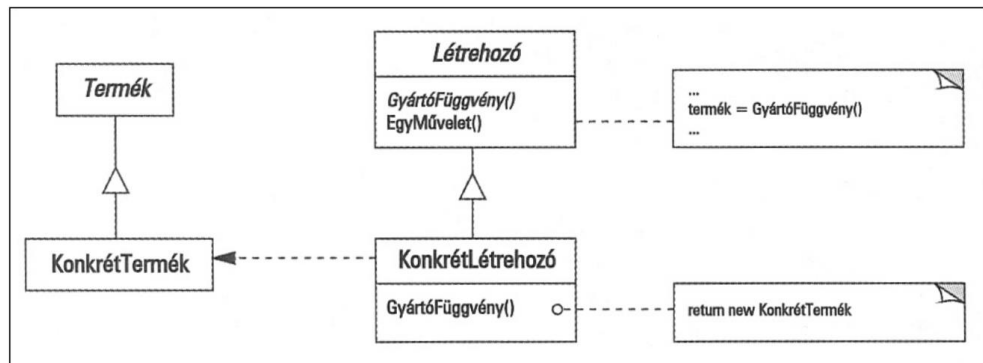
Alkalmazhatóság

Az osztály nem tudja előre megjósolni, milyen objektumosztályt kell létrehoznia.

Az osztály azt szeretné elérni, hogy az alosztályai adják meg, milyen objektumot kell létrehozni.

Az osztályok átruházzák a felelősséget a számos segítő alosztály egyikére és mi szeretnénk tudni, melyik lett a képviselő.

Szerkezet



Prototípus (Prototype) – Objektum-létrehozási minta

Cél

Prototípus példány használatával meghatározni, milyen típusú objektumokat kell létrehozni, az új objektumokat pedig ennek a prototípusnak a lemásolásával előállítani.

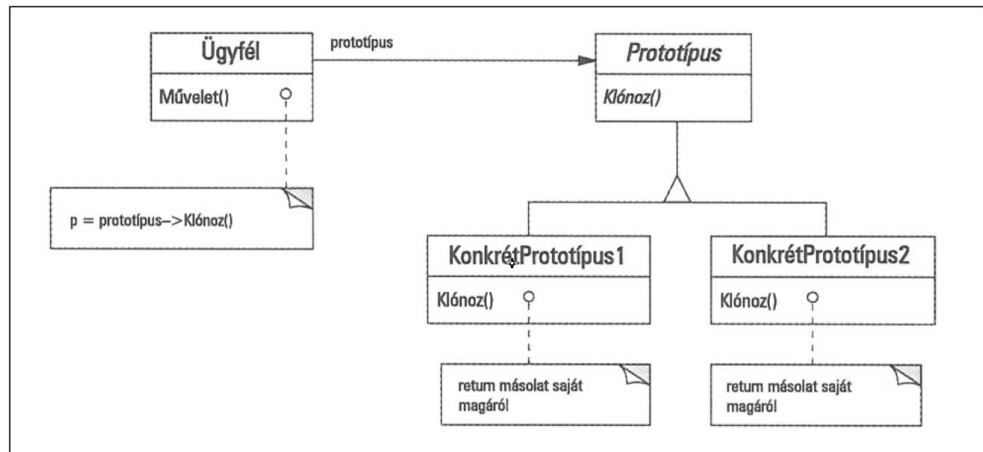
Alkalmazhatóság

A Prototípus mintát akkor használjuk, amikor a rendszernek függetlennek kell lennie a termékek létrehozásának, összeállításának és megjelenítésének módjától és

- amikor a példányosítandó osztályokat futásidőben adják meg, például dinamikus betöltés útján, vagy

- ha el szeretnénk kerülni a termékekével párhuzamos osztályhierarchiájú gyárak építését, vagy
- amikor egy osztály példányainak csak néhány különböző állapotkombinációja jöhet létre. Kényelmesebb lehet a megfelelő számú prototípust beépíteni, majd klónozni, mint egyenként példányosítani az osztályokat minden alkalommal a megfelelő állapottal.

Szerkezet



Egyke (Singleton) – objektum-létrehozási minta

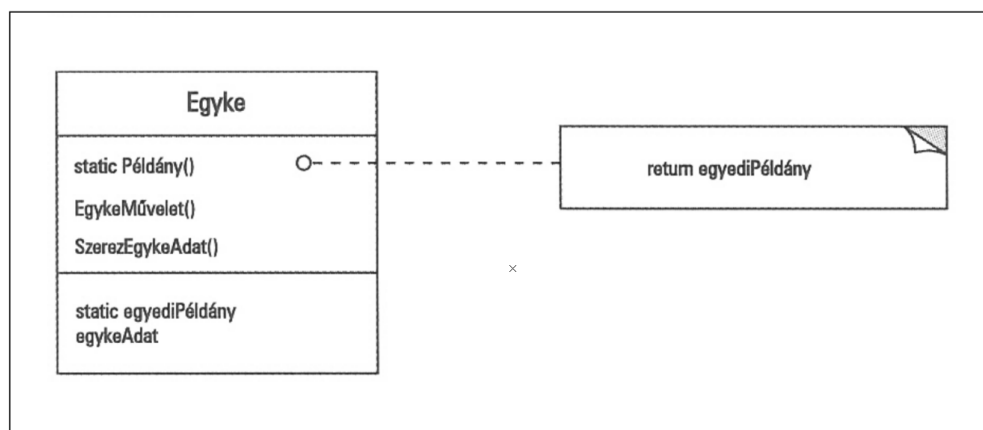
Cél

Egy osztályból csak egy példányt engedélyezni és ehhez globális hozzáférési pontot megadni

Alkalmazhatóság

- Pontosan egy példányra van szükség valamelyik osztályból és annak elérhetőnek kell lenni az ügyfelek számára a jól ismert elérési pontokból.
- Ennek az egyetlen példánynak alosztályokkal bővíthetőnek kell lenni és az ügyfeleknek képeseknek kell lenni a saját kódjuk módosítása nélkül használni a bővített példányt.

Szerkezet



Felhasznált irodalom

Erich Gamm, Richard Helm, Ralph Johnson, John Vlissides

Programtervezési minták – Újrahasznosítható elemek objektumközpontú programokhoz

2004 Kiskapu Kiadó

Borítókép:

<https://gpcoder.com/wp-content/uploads/2018/08/design-patterns.jpg>