

Gang Of Four (GoF) minták

GoF névvel egy könyv négy szerzőjét szokták illetni, akik a könyvben a szoftvertervezés jellemző problémáira adnak ismétlődő válaszokat. A könyvük első felében az objektumorientált programozás lehetőségeit és csapdáit veszik át, a második részben a 23 klasszikus szoftver tervezési mintát.

[Erich Gamma](#), [Richard Helm](#), [Ralph Johnson](#) and [John Vlissides](#)

- Jellemzően C++ és Smalltalk példákat használnak. Létrehozó: (creational)
 - A minták általában a megadott osztályok további kidolgozásával és példányosítással, illetve példányosításra használhatók...
- Szerkezeti: (structural)
 - A minták objektumösszetételekre épülnek
- Viselkedési (behavioral):
 - A minták a megadott osztályok példányai közötti kommunikációra épülnek

GoF létrehozó minták

- Singleton: egypéldányú osztályok létrehozására
- Prototype: létező objektumok lemásolására
- Builder: összetett szerkezetek létrehozására
- Factory Method:
- AbstractFactory: közös témához köthető objektumok létrehozására szolgál

Singleton

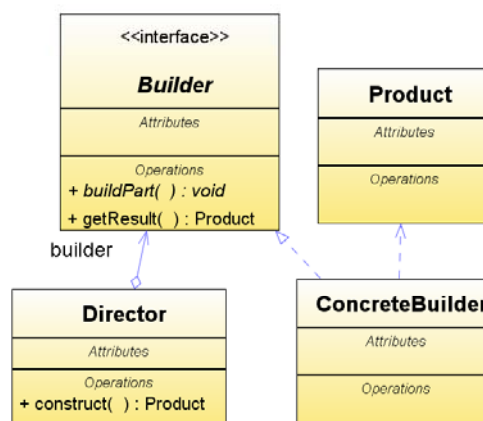
- Olyan osztály megadása, amely megakadályozza, hogy többszörösen példányosítsák.
- Ezt úgy éri el, hogy az osztály tartalmazza saját egyetlen példányát (statikus, osztály hatókörű)
- Az egyetlen példány elérése statikus (osztály hatókörű) elérő (accessor) művelettel
- A Singleton konstruktora csak privát lehet! (hogy ne tudjok bárki új példányt létrehozni belőle)
- Késői kezdőértékezés
- Vigyázat a statikus accessorral! Nem származtatható le belőle újabb osztály
- Globális változóhoz hasonló működésmód (állandónak látszó környezeti információk tárolása vagy csak funkcionális osztály)
- Singleton törlése... (?)

Singleton	
<i>Attributes</i>	
-	uniqueInstance : Singleton
-	singletonData : int
<i>Operations</i>	
+	instance() : Singleton
+	singletonOperation() : void
+	getSingletonData() : void

- **Létrehozói minták: objektumok létrehozásnak módjával foglalkozik, hogy milyen módon hozzuk létre**
- az objektumokat a helyzetnek megfelelő módon.
- o Singleton: olyan objektum, amiből garantáltan egyetlen példány létezhet csak egy rendszerben. Rendszerint tartalmazza saját egyetlen példányát.
- o Abstract factory: egy feladathoz szükséges objektumot nem a saját konstruktorával hozzuk létre, hanem a factory egy függvényhívásával. Egy felületet ad akár összetartozó, akár független objektumok létrehozásához, anélkül, hogy megadnánk a konkrét osztálytípust.
- o Builder: elválasztja egy komplex objektum létrehozását a reprezentációjától, így ugyanaz a létrehozó eljárás képes különböző reprezentációkat is létrehozni.
- o Prototype: megad egy objektum prototípust, amiből aztán másolással lehet létrehozni újabb példányokat.

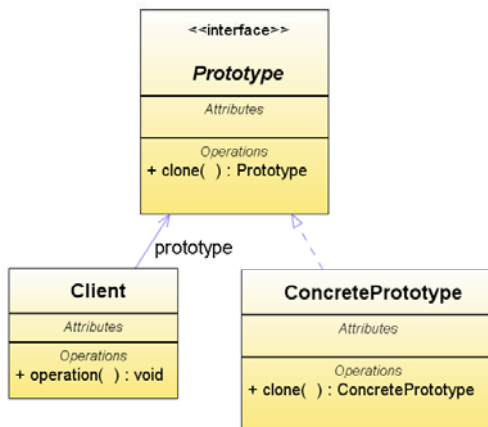
Builder

- Cél: valamilyen külső reprezentáció beolvasásakor a reprezentáció felismerését és az adatszerkezet létrehozását különválasztani
- Megoldás: a Director a Builder burkolója (Builder része Directornak).
- A Director:construct eljárása létrehozza a teljes Product szerkezetet.
- A construct eljárás meghívja a Builder:buildPart1(), ...buildPartN() eljárásait, amelyek a Builderen belül létrehozzák a Product részeit.
- A létrehozás után a Builder:getResult:Product eljárással lehet az eredményt megkapni.



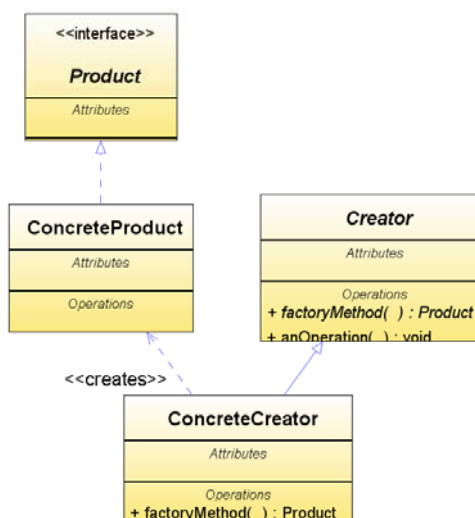
Prototype

A Prototype típusú objektumból új példányokat úgy hozhatunk létre, hogy meghívjuk a `clone():Prototype` módszert. Ez létrehozza az alapobjektum egy másolatát.



Factory Method

A munkához szükséges objektumokat a `ConcreteCreator:factoryMethod` állítja elő. Ennek az eredménye az absztrakt `Product` típus. A változatlan specifikáció mellett egy vagy több `ConcreteProduct` létezik, a `factoryMethod` természetesen konkrét típusú objektumokat állít elő.



AbstractFactory

A munkához szükséges objektumokat a programban nem a new operátorral állítjuk elő, hanem a Factory osztály createProduct (polimorf) hívásával.

A Factory osztályban nemcsak egyetlen createProduct eljárás van, hanem ilyenekből egy csokorra való.

Ha új ConcreteProduct típusra van szükségünk, akkor nem szükséges az absztrakt kódot módosítani, hanem csak a megfelelő ConcreteFactory osztályt kell létrehozni.

Ilyen módon hozzuk létre pl. a CORBA objektumokat (a távoli szerverről csak a Factory objektum azonosítóját szerezzük meg, a többit már a megfelelő hívások eredményeképpen kapjuk meg).

