

Yaratımsal Kalıplar III

Yapıcı (Builder)

- Tek ara yüz kullanarak karmaşık bir nesne grubundan gerektiğinde parça yaratılmasını sağlar. Nesne grubu kullanıldıkça istenilen şekilde yapılır ve bu sayede kullanılmayan parçaların gereksiz yere yaratılarak kaynak harcama durumu ortadan kaldırılmış olur.

- Ürettiğimiz yapılandırıcı metod bize gereksiz gelen parametrelerden (uzun yapılandırıcı anti patterni) oluşabilir. Bu sıkıntıdan kaçınmak için nesnemizin alacağı parametrelere göre değişkenlik gösteren metod oluşturma yöntemidir.
- Varsayılan yapılandırıcı ve set metotları (JavaBean), uzun yapılandırıcı problemini başka bir probleme evirmektedir. Çok sayıda set metodu çağırılması hataya sebep olabilmektedir. Inner class ile bu sıkıntının önüne geçilebilir. Böylece nesnemiz immutable'da olmuş olur.

- Builder kalıbında genellikle iç sınıf olarak static bir sınıf kullanılır. Bu, Builder sınıfının örneklerinin, oluşturulacak nesne türünden bağımsız olarak örneklenmesine olanak tanır.
- Bu şekilde, Builder sınıfı, oluşturulacak nesne türüne özgü ayrıntılar içeren özellikleri alabilir ve ilgili özelliklerin set edilmesi için yöntemler sağlayabilir. Bu sayede, oluşturulacak nesne türünden bağımsız olarak, Builder sınıfı, herhangi bir nesne için özelleştirilebilir.
- Inner (iç) sınıf kullanılması, Builder nesnesi oluşturulurken gereksiz nesne yaratma maliyetlerini önler ve Builder nesnesinin oluşturulmasıyla ilgili karmaşıklığı en aza indirir. Bu sayede, oluşturulacak nesnenin özellikleri belirli bir sıraya göre ayarlanabileceği, eksik veya yanlış özelliklerin engellenebilir ve nesnenin tutarlı bir şekilde oluşturulması sağlanır.
- Ayrıca, iç sınıf olarak static bir sınıf kullanmak, kodun daha okunaklı ve düzenli hale gelmesine de yardımcı olabilir. Builder sınıfının içinde, Builder nesnesini oluşturmak için kullanılan yöntemler açıkça belirtilir ve bu sayede, Builder nesnesinin kullanımı daha anlaşılır hale gelir.

- Builder kalıbı Interface kullanarak da yapılabilir. Araba nesnesini oluşturmak için Builder arayüzünü kullanır. Araba sınıfı, Builder arayüzüne sahip değildir; bunun yerine, Builder arayüzü, Builder sınıfı tarafından uygulanır. Bu, Araba sınıfının özelliklerine erişim hakkını Builder sınıfına vermez, böylece Araba nesnesi sadece oluşturulduktan sonra özelliklerinin değiştirilmesini engeller. Bu, özellikle Araba sınıfının değişmez (immutable) olmasını sağlamak istediğimiz durumlarda faydalıdır.
- Builder tasarım kalıbı, nesne oluşturma sürecini karmaşıktırmadan ve daha esnek bir yapıda tasarlamamıza olanak sağlar. Ayrıca, nesne özellikleri değişken olduğunda, nesne oluşturma işlemini daha kolay hale getirir.

- Bu kalıbın amacı, karmaşık nesne oluşturma sürecini yönetmektir. Factory Metot ve Abstract Factory kalıpları sadece nesne oluşturma sorumluluğunu devralmak içindir.

Kaynakça

- <https://en.wikipedia.org/>
- <https://bidb.itu.edu.tr/>
- <https://www.tutorialspoint.com/>
- <https://medium.com>
- [Javaturk.org](https://javaturk.org)
- <https://blog.burakkutbay.com/>