**Memento (Muhafız) Kalıbı (Pattern)**

• **Intent**: Without violating encapsulation, capture and externalize an object's internal state so that the object can be restored to this state later.

• **Amaç**: Sarmalamayı bozmadan, bir nesnenin iç durumunu yakala ve dışarıya çıkar öyle ki nesne bu duruma daha sonra dönebilsin.

**PROBLEMS**

• Bazen bir nesnenin durumunu kaydetme ihtiyacı duyulur.

• Özellikle bir nesnenin durumu sıklıkla değişiyor ve zaman zaman bu değişiklikleri geri almak (**undo**) ya da geçmişteki bir noktaya dönmek (**checkpoint**) söz konusu ise, bu ihtiyaç çok daha açık hale gelir.



• Sarmalama (**encapsulation**) nesnenin durumunu dışarıya açmasına izin vermediğine göre, bir nesnenin durumunu nasıl dışarıya kaydedebiliriz?

**• GoF, grafik editör örneğini verir.**

• Grafik editör farklı türde grafik nesneleri ve aralarındaki bağlantıları tutmaktadır.

• Böyle bir sistemde geri alma (undo) işlemini desteklemek için grafik nesneler ile aralarındaki ilişkilerin, koordinat, uzunluk vb. Türden durumlarını saklamak gereklidir.

• Böyle bir ihtiyacı, durumu saklanacak nesnenin durumuna ulaşıp, saklayan ve sonra gerektiğinde geri veren bir başka nesne ile yapmak düşünülebilir.

• Ama bu çözümde istenmeyen şey durumu saklanacak nesnenin sarmalanmış (**encapsulated**) yapısının korunmasıdır.

**SOLUTION**

• Bu problem **Memento** (hatıra(t), muhafız) ile çözülebilir.

• **Memento** bir nesnedir ve durumu saklanacak nesnenin bir andaki (snapshot) durumu için muhafızlık yapar.

• **Memento**, gerektiğinde durumu değişen nesnenin durumunu, sakladığı hale geri getirmekte kullanılır.

• **Memento**, nesnenin sadece durumu saklanma ihtiyacı duyulan kısmını gereken derinlikte saklar.

• Çözümde **Originator**, durumu saklanacak olan nesnedir.

• **Memento**, **Originator’ın** durumunu saklayan nesnedir.

• **Caretaker**, **Originator’ın** durumunu saklaması için Memento’yu tetikleyen nesnedir.



**• Caretaker, Memento’nun save() ve undo() arayüzünü bilir.**

• Bunun dışında **Originator’ın** durumuna ulaşmaya çalışmaz.

• Birden fazla farklı tipte **Caretaker**, olabilir ve her biri farklı sebeplerle **Memento’nun** save**()** ve **undo**() metotlarını tetikleyebilir.

• **Memento,** **Originator’ın** saklamaya ihtiyaç duyduğu duruma ulaşacak arayüzünü bilir.

**SONUCLAR**

• **Memento** kalıbı ile durumu saklanan nesnenin sarmalaması (**encapsulation**) bozulmaz.

• Fakat, **Memento’nun** tutacağı nesne durumunun miktarı yükseldikçe, bu kalıbı uygulamak pahalı hale gelebilir.

• Bu durumda **artımlı** (**incremental**) **Memento** kullanmak bir çözüm olabilir.

• **Tarihselliğin** gerekli olduğu hallerde, çok daha karmaşık **Memento** nesneleri kullanılabilir.

• **Memento** kalıbı ile durumu saklanan nesne daha basit hale gelir, aksi tak tirde nesne kendi tarihselliğini tutması gerekir.

• Bu şekilde durumu saklanan nesne **SRP** açısından daha iyi durumda olur

**KULLANIMLAR**

**• Geri alma (undo) işlemlerinin çok sık kullanıldığı GUI, kelime işlemci**

**(editor) vb. sistemlerde Memento kalıbı kullanır.**

**DİĞER KALIPLARLA İLİSKİ**

• **Command** kalıbı, “durumu sakla” ve “geri al” (undo) işleri için kullanılabilir.

• **Memento** nesnesinin tuttuğu durumlar için **Iterator** kalıbı kullanılabilir

UYGULAMA

**Account** nesnesinin tarihselliğini sağlamak amacıyla Memento

kalıbından nasıl faydalanırsınız?