

Nesne Yönelimli Analiz ve Tasarım

Nesne Tasarımı

Tasarım Desenleri
Design Patterns

Tasarım Desenleri

- * Yazılımlar geliştirilirken karşılaşılan genel tasarım problemlerini tanımlayarak, bu problemin en uygun nasıl çözülebileceğini (kod tekrar kullanımını artırmak ve değişikliği kolaylaştırmak için) ve çözümün ortaya çıkaracağı sonuçları anlatır.
- * Programlama dillerinden bağımsızdır.
- * Yazılım geliştiricilerin tasarımlarla ilgili tartışma yapmasını kolaylaştırır.
- * Tasarım desenleri dört bölümden oluşur
 - * Desenin adı
 - * problemin tanımı: desenin ne zaman/herede kullanılabileceği
 - * çözüm: problemin nasıl çözüleceği (kullanılması gereken bileşenler, aralarındaki bağıntı vb.)
 - * sonuç: deseni uygulamanın sonuçları?

Tasarım Desenleri

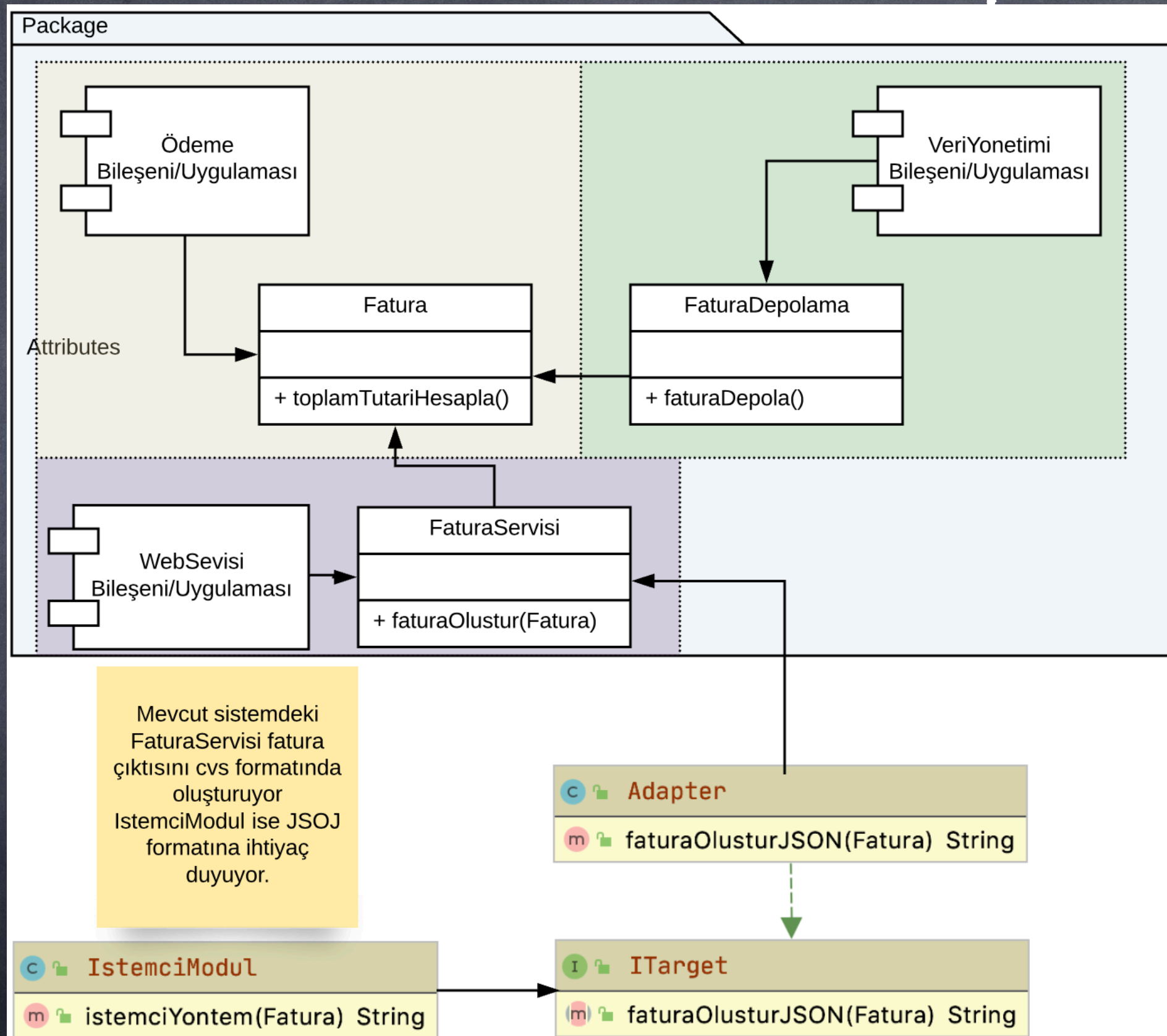
- * Creational(Nesne oluşturma): Nesnelerin uygun bir şekilde oluşturulmasını sağlayacak mekanizmalar içerir.
- * Structural(Yapısal): Nesnelerin sistemler içerisinde uygun olarak yerleştirilmesini sağlayacak desenlerdir.
- * Behavioral(Davranışsal): Nesnelerin birbirleriyle uygun olarak etkileşiminini düzenleyen mekanizmalar.

Creational (Nesne oluşturma)	Structural (Yapısal)	Behavioral (Davranışsal)
Singleton Pattern	Adapter Pattern	Template Method Pattern
Factory Pattern	Composite Pattern	Mediator Pattern
Abstract Factory Pattern	Proxy Pattern	Chain of Responsibility Pattern
Builder Pattern	Flyweight Pattern	Observer Pattern
Prototype Pattern	Facade Pattern	Strategy Pattern
	Bridge Pattern	Command Pattern
	Decorator Pattern	State Pattern
		Visitor Pattern
		Interpreter Pattern
		Iterator Pattern
		Memento Pattern

Tasarım Desenleri : Adapter

- * Yapısal desenlerden biridir.
- * Bir sınıfın arayüzünü istemci tarafından beklenen başka bir arayüze dönüştürmek için kullanılır.
- * Mevcut sınıfların (kaynak kodunu değiştirmeden) diğer sınıflarla çalışabilmesini sağlar.
- * Bu desen sayesinde, daha önceden geliştirilmiş modülleri, yeni modüller ile birlikte (bu modüllere uygun arayüzlere sahip olmadığı durumlarda) kullanabiliriz.

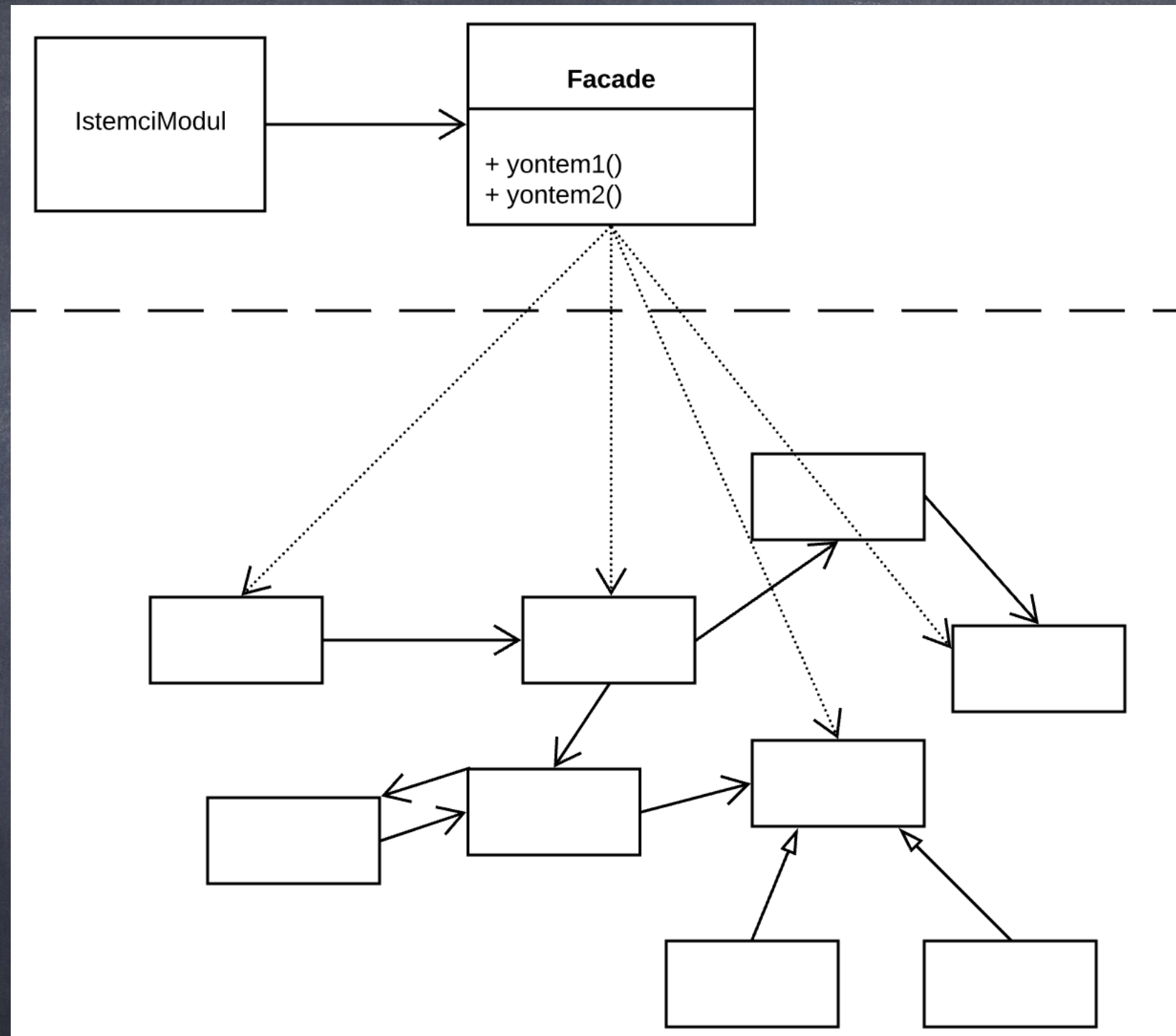
Tasarım Desenleri : Adapter



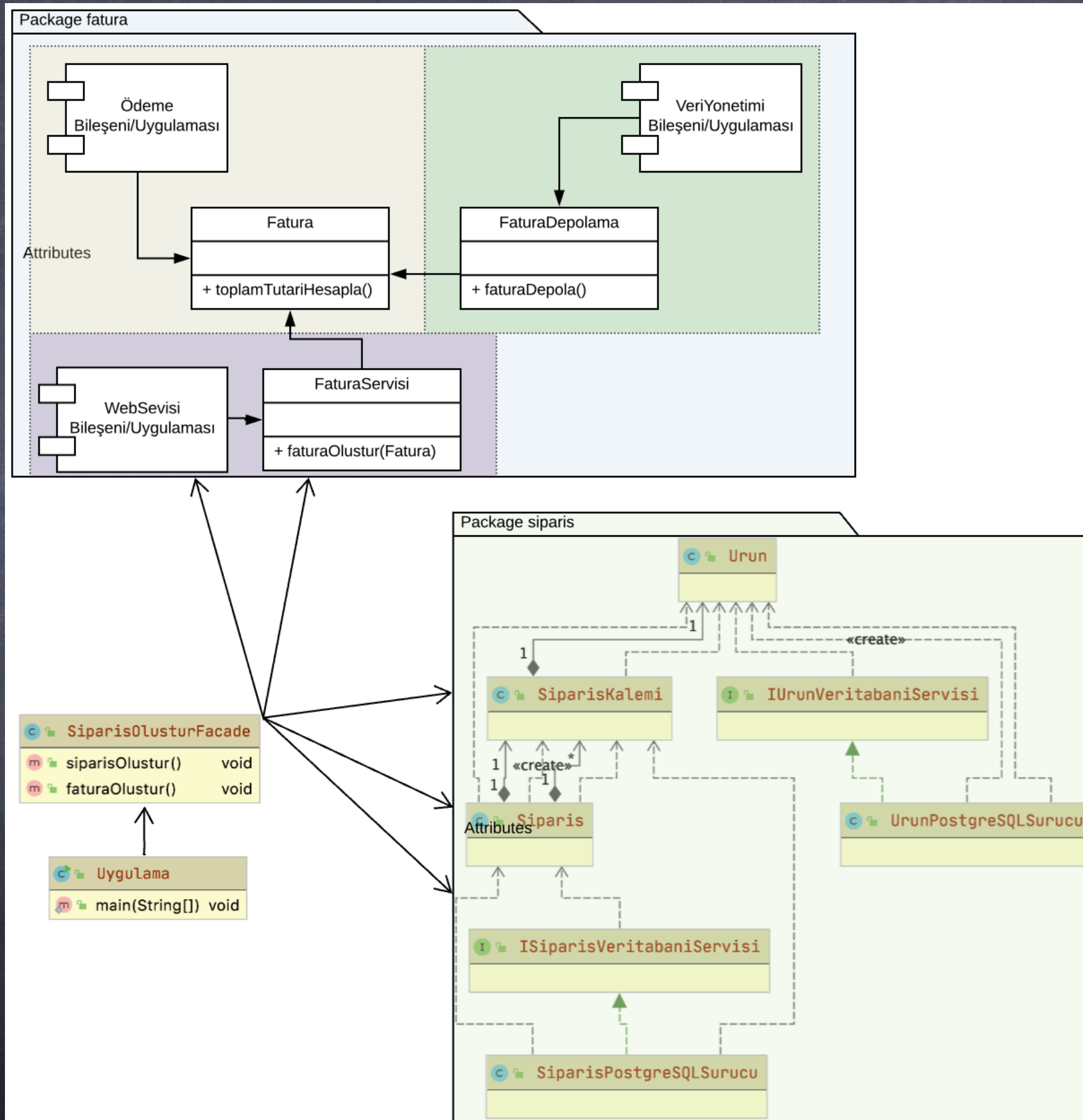
Tasarım Desenleri : Facade

- * Yapısal desenlerden biridir.
- * Karmaşık yapıdaki bir sınıf topluluğu (kütüphane, alt bileşen, eskiden yazılmış kodlar vb.) için basitleştirilmiş arayüz sağlar.
- * İstemci kodun karmaşık alt sistemle etkileşimini kolaylaştırır (loosly coupling).
- * İyileştirilme imkanı olmayan eski modüllerle etkileşim imkanı sağlar.
- * Anlaşılabilirliği artırır.
- * Yazılım içerisinde katmanlar oluşturulmasına imkan verir.

Tasarım Desenleri : Facade



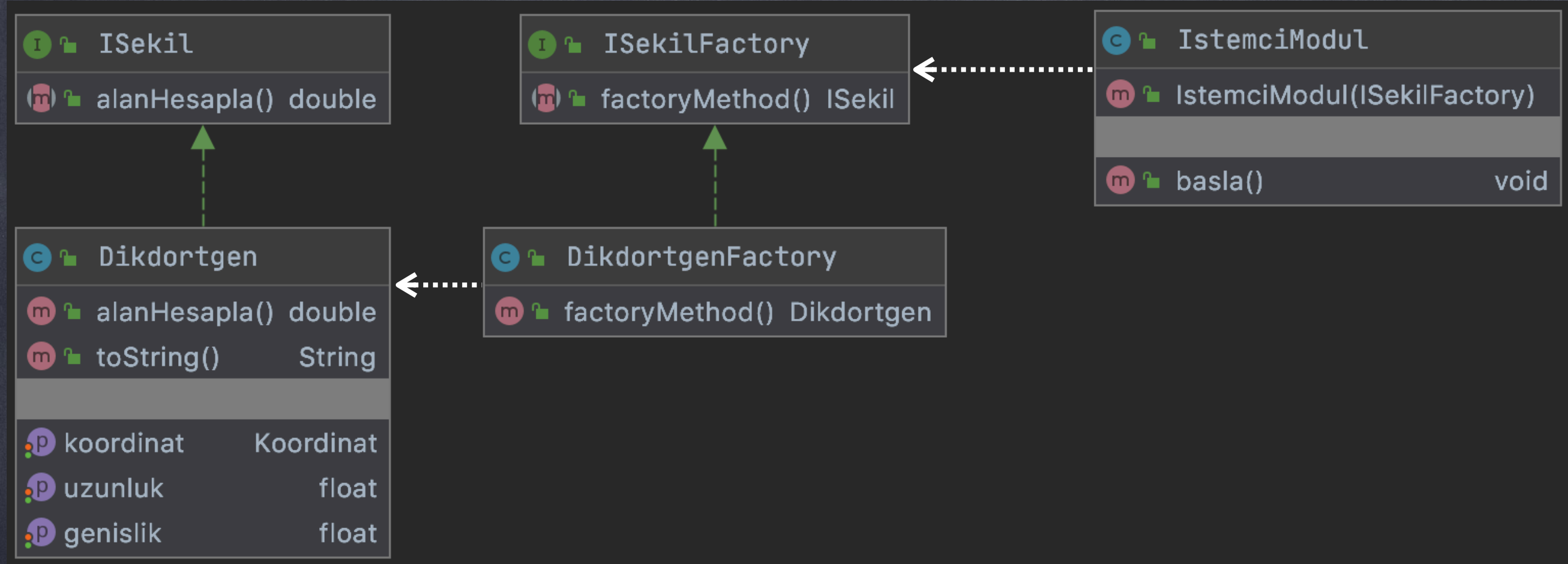
Tasarım Desenleri : Facade



Tasarım Desenleri-Factory Method

- * Nesne oluşturmaya (creational) ilgili desenlerden biridir.
- * Bir sınıftan nesne oluşturmak gerektiğinde, bu sorumluluğu istemci koddan ayırmak (kapsülleme/SRP) için kullanılır.
- * Özellikle çok sayıda parametre göndermek gerektiğinde (kalıtım söz konusu ise) nesne oluşturma işi karmaşıklaşır (kod tekrarı, değişikliklerden istemci kodun etkilenmesi, kodların kötü görünmesi...).
- * Nesne oluşturmak gerektiğinde "factory method" çağırılarak nesne oluşturulur. Sınıfın bulunduğu yol/paket, istisna yönetimi v.s. uğraşmaya gerek yok. Nesne oluşturmaya ilgili herhangi bir değişiklikten (yolların değişimi, parametre değişimleri v.s.), istemci kod etkilenmez.

Tasarım Desenleri-Factory Method (uygulama1)



Tasarım Desenleri-Factory Method (uygulama2)

