

Yazılım Tasarım Desenleri Nedir?

Çalışma Kağıdı

Yazılım tasarım desenleri, belirli bir bağlamda ortaya çıkan ve sıkça tekrarlanan tasarım sorunlarına yönelik, genel bir çözüm tarifidir. Bu, kodun yeniden kullanılabilirliğini ve bakımını kolaylaştırır.

Sorular

1. Bir sınıfın birden fazla örneğinin oluşturulmasını engellemek için hangi tasarım deseni kullanılır?
A) Factory Method
B) Observer
C) Singleton
D) Adapter
2. Nesne oluşturma mantığını soyutlamak ve alt sınıfların hangi nesneyi oluşturacağına karar vermesini sağlamak için hangi desen kullanılır?
A) Abstract Factory
B) Factory Method
C) Builder
D) Prototype
3. Bir nesnenin durumundaki değişiklikleri diğer nesnelere otomatik olarak bildiren desen hangisidir?
A) Strategy
B) Observer
C) Mediator
D) Command
4. Bir 'Singleton' tasarım deseni ne zaman kullanılır?
5. 'Factory Method' deseni hangi problemi çözer?
6. 'Observer' deseni ne işe yarar?
7. Tanımla: Yazılım Tasarım Desenlerinin Temel Amacı Nedir?
8. Tanımla: En Sık Kullanılan Tasarım Desenleri Kategorileri Nelerdir?
9. Tanımla: Singleton Deseni Ne İçin Kullanılır?

Cevap Anahtarı

1. C) Singleton — Singleton deseni, bir sınıfın yalnızca bir örneğinin oluşturulmasını garanti eder.
2. B) Factory Method — Factory Method deseni, nesne oluşturma sorumluluğunu alt sınıflara devreder.
3. B) Observer — Observer deseni, bir özne ile ona bağlı gözlemciler arasında bir 'birçok' bağımlılık ilişkisi tanımlar.
4. Singleton deseni, bir sınıfın yalnızca bir örneğinin olmasını ve bu örneğe global bir erişim noktası sağlamak istediğimiz durumlarda kullanılır. Örneğin, bir veritabanı bağlantısı veya yapılandırma yöneticisi gibi kaynaklar için tek bir örnek yeterli olduğunda tercih edilebilir.
5. Factory Method deseni, nesne oluşturma işlemini soyutlayarak, hangi sınıfın hangi nesneyi oluşturacağına alt sınıfların karar vermesini sağlar. Bu, kodun daha esnek olmasını ve yeni nesne türleri eklendiğinde ana kodu değiştirmeye gerek kalmamasını sağlar.
6. Observer deseni, bir nesnenin (özne) durumundaki değişiklikleri, ona bağımlı olan diğer nesnelere (gözlemciler) otomatik olarak bildirmesini ve güncellemesini sağlar. Bu, nesneler arasında gevşek bağıllık kurar.
7. Tekrarlayan tasarım problemlerine genel ve kanıtlanmış çözümler sunmak.
8. Yapısal (Structural), Davranışsal (Behavioral) ve Oluşturma (Creational) desenleri.
9. Bir sınıfın yalnızca bir örneğinin olmasını sağlamak ve bu örneğe global erişim sunmak için.

Bounlu

Tüm kartlar, adım adım çözümler ve AI hoca desteği Notek uygulamasında.
Sınav tarihlerini Promy otomatik hatırlatıcıya çevirir.