

Nesne Yönelimli Tasarım İlkeleri Nedir?

Çalışma Kağıdı

Nesne Yönelimli Tasarım ilkeleri, yazılımın modülerliğini, esnekliğini ve bakımını artırmak için kullanılan bir dizi kılavuzdur ve genellikle SOLID prensipleri ile özetlenir.

Sorular

1. Hangi prensip, bir sınıfın yalnızca tek bir işlevi yerine getirmesi gerektiğini savunur?
A) Açık/Kapalı Prensibi
B) Tek Sorumluluk Prensibi
C) Liskov Yerine Geçme Prensibi
D) Arayüz Ayırma Prensibi
2. Bir türetilmiş sınıf, temel sınıfın yerine kullanıldığında programın doğruluğu bozulmuyorsa, hangi prensip sağlanmış olur?
A) Arayüz Ayırma Prensibi
B) Bağımlılıkların Tersine Çevrilmesi Prensibi
C) Liskov Yerine Geçme Prensibi
D) Tek Sorumluluk Prensibi
3. Yazılım varlıklarının genişletmeye açık, ancak değiştirmeye kapalı olması gerektiğini savunan prensip hangisidir?
A) Tek Sorumluluk Prensibi
B) Bağımlılıkların Tersine Çevrilmesi Prensibi
C) Liskov Yerine Geçme Prensibi
D) Açık/Kapalı Prensibi
4. Tek Sorumluluk Prensibi (SRP) ne anlama gelir?
5. Açık/Kapalı Prensibi (OCP) ne anlama gelir?
6. Liskov Yerine Geçme Prensibi (LSP) ne anlama gelir?
7. Tanımla: Tek Sorumluluk Prensibi (SRP)
8. Tanımla: Açık/Kapalı Prensibi (OCP)
9. Tanımla: Liskov Yerine Geçme Prensibi (LSP)

Cevap Anahtarı

1. B) Tek Sorumluluk Prensibi — Tek Sorumluluk Prensibi (SRP), bir sınıfın yalnızca tek bir sorumluluğu olması gerektiğini belirtir.
2. C) Liskov Yerine Geçme Prensibi — Liskov Yerine Geçme Prensibi (LSP), türetilmiş sınıfların temel sınıflarının yerine geçebilmesini garanti eder.
3. D) Açık/Kapalı Prensibi — Açık/Kapalı Prensibi (OCP), mevcut kodu değiştirmeden yeni özellikler eklemeyi teşvik eder.
4. Bir sınıfın yalnızca tek bir sorumluluğu olmalı, yani yalnızca tek bir işlevi yerine getirmelidir. Bu, sınıfın neden değişebileceği için yalnızca tek bir nedeni olmasını sağlar.
5. Yazılım varlıkları (sınıflar, modüller, fonksiyonlar vb.) genişletmeye açık, ancak değiştirmeye kapalı olmalıdır. Yeni özellikler eklemek için mevcut kodu değiştirmek yerine, kodu genişletmek tercih edilmelidir.
6. Türetilmiş sınıflar, temel sınıflarının yerine geçebilmelidir. Yani, bir temel sınıfın nesnelerinin beklendiği yerde, onun türetilmiş sınıfından bir nesne kullanıldığında programın doğruluğu bozulmamalıdır.
7. Bir sınıfın yalnızca bir sorumluluğu olmalı.
8. Genişletmeye açık, değiştirmeye kapalı olmalı.
9. Türetilmiş sınıflar temel sınıflarının yerine geçebilmeli.

Bounlu

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