Çalışma zamanı hatası nedir?



www.yzokul.com.tr

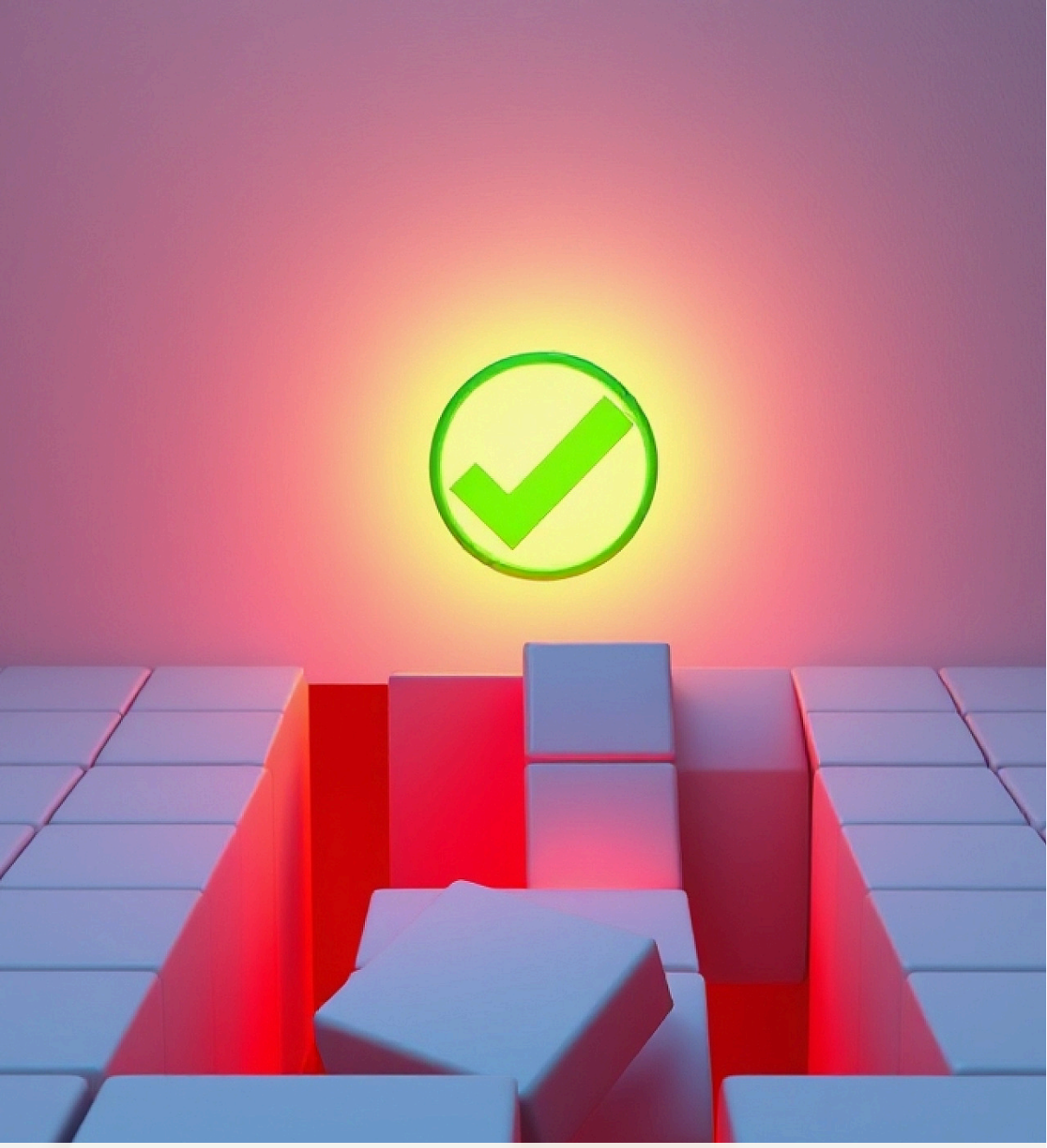
Çalışma zamanı hatası nedir?



Şunu soruyorsun: çalışma zamanı hatası nedir? Kısa yanıt, program yazılırken değil, gerçekten çalışırken ortaya çıkan bir sorun olduğudur. Bu, hatanın bilgisayar talimatları izlerken, yürütme sırasında ortaya çıktığı anlamına gelir. Yazılı metindeki bir hatadan farklıdır. Bu ayrımı anlamak, bu tür hatalarla başa çıkmanın ilk adımıdır.



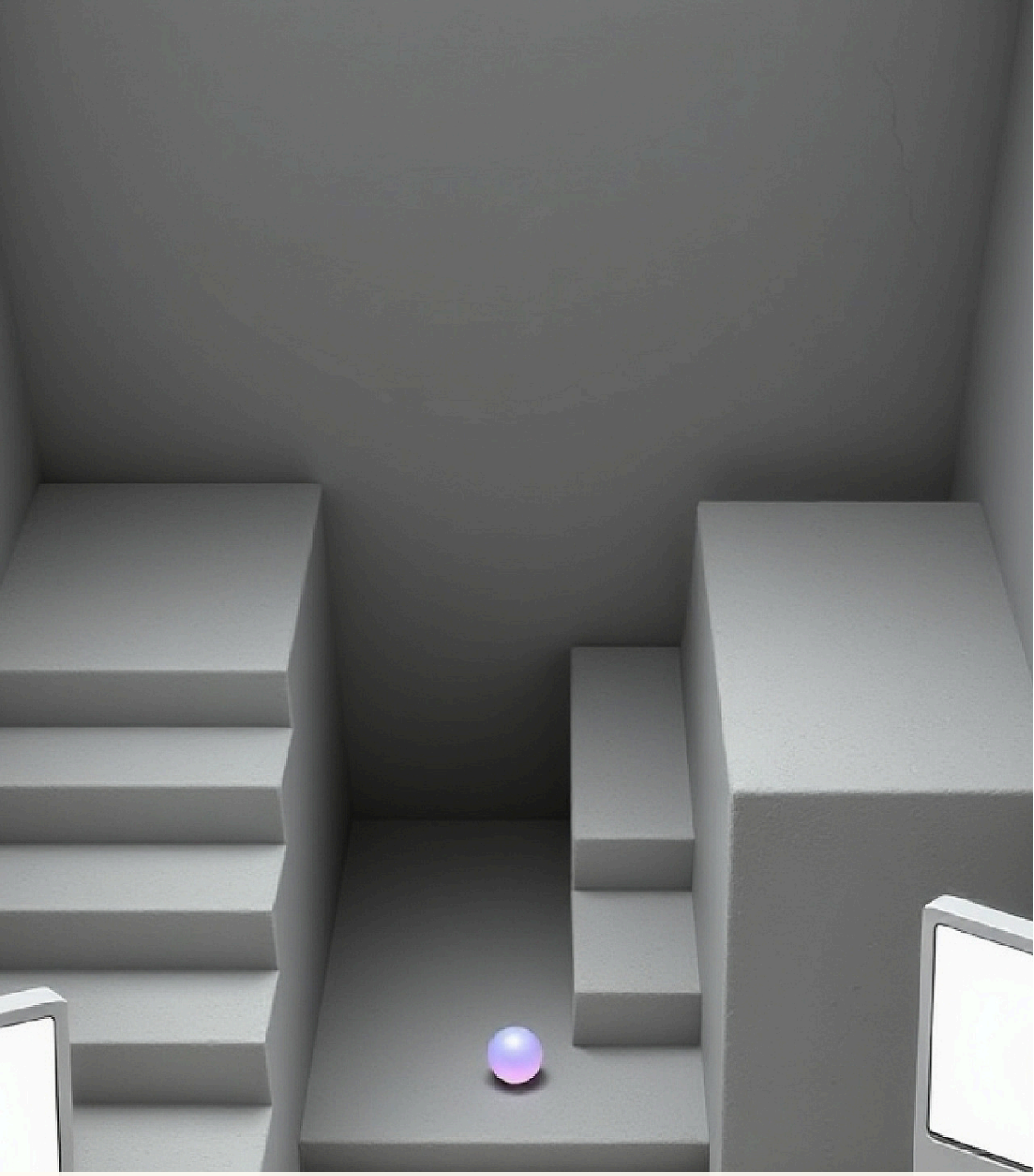
Bir program önce metin olarak yazılır, sonra bilgisayar tarafından çalıştırılır. Bunlar iki ayrı aşamadır: yazma ve çalıştırma. Bir hata bu aşamalardan herhangi birine ait olabilir. Yazma hataları, metin programlama dilinin kurallarını ihlal ettiğinde ortaya çıkar. Çalışma zamanı hataları ise daha sonra, bilgisayar talimatları yürüttüğünde, çalıştırma aşamasında ortaya çıkar. Bir hatanın hangi aşamadan geldiğini bilmen gerekir.



Kod tüm yazma kurallarına uyabilir ve yine de daha sonra başarısız olabilir. Yazma kuralları yalnızca metni denetler, çalıştığında ne olduğunu değil. Örneğin, bir program dile göre kusursuz yazılmış olabilir, ancak çalıştığında var olmayan bir dosyayı açmaya çalışır. Metin doğrudur, ancak çalışma zamanındaki durum geçersizdir. Bu bir çalışma zamanı hatasıdır.



Program çalışırken bilgisayarın talimatları tek tek izlediğini düşün. Bir çalışma zamanı hatası, bu talimatları izleme sırasında ortaya çıkar. O anda geçersiz bir şey olduğu için normal akışı kesintiye uğratır. Örneğin, program bir sayıyı sıfıra bölmeye çalışabilir, ki buna izin verilmez. Bilgisayar normal şekilde devam edemez, bu yüzden bir hatayla durur.



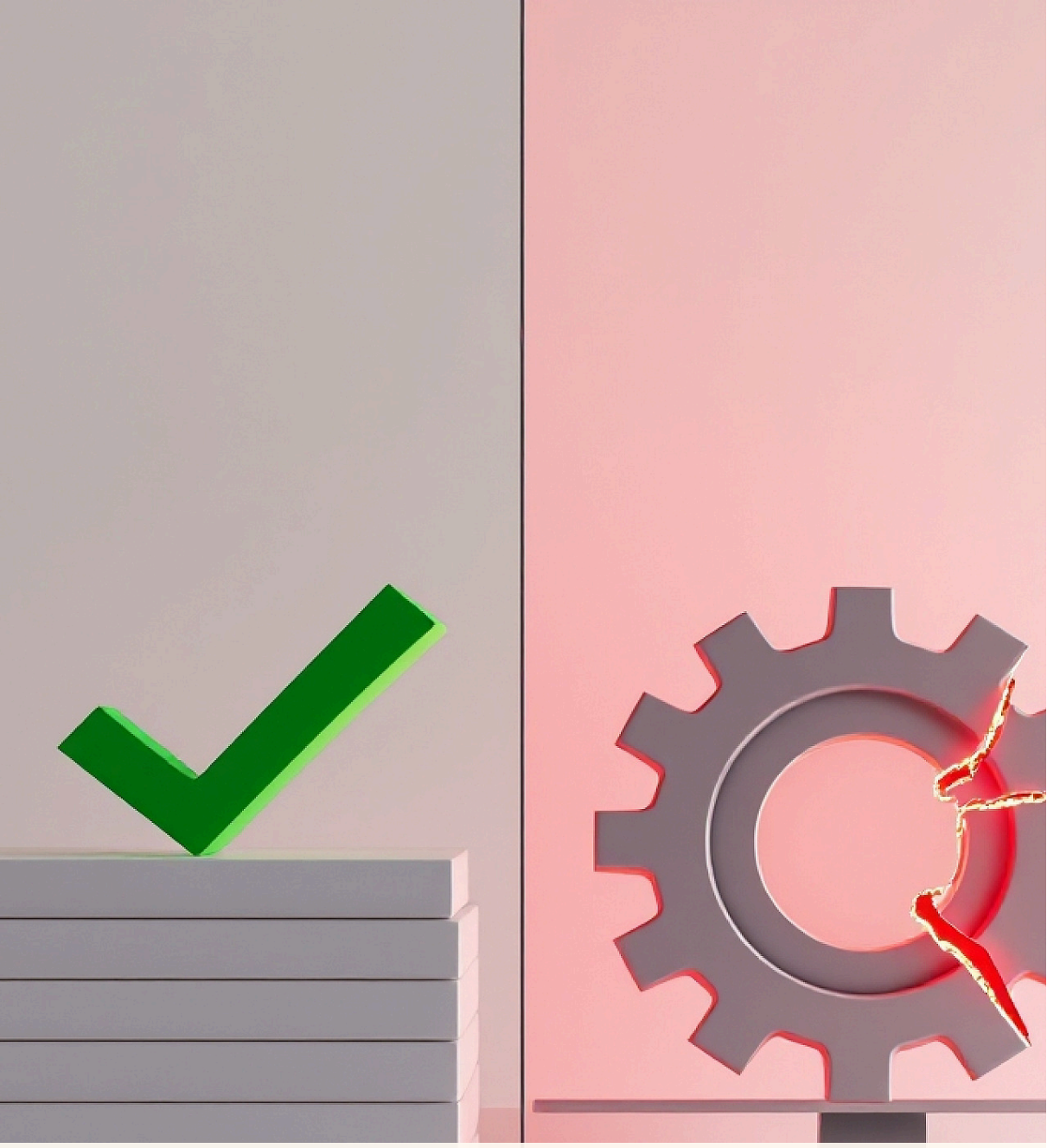
Bir programı adımlar listesi olarak düşün; ortadaki bir adım yanlış gider. Hata, o anda normal akışı durdurur veya kesintiye uğratır. Örneğin, üçüncü adım bir dosyadan veri okumak olabilir, ancak dosya orada değildir. Program bu adımı tamamlayamaz, bu yüzden durur ve bir çalışma zamanı hatası bildirir. Önceki ve sonraki adımlara ulaşılmaz.



Var olmayan bir dosyayı açmaya çalışan bir program düşün. Kod doğru yazılmıştır, ancak çalışma zamanındaki durum geçersizdir. Dosya silinmiş veya hiç oluşturulmamış olabilir. Program çalıştığında işletim sisteminden dosyayı ister ve sistem dosyanın bulunamadığını söyler. Beklenen dosya eksik olduğu için bu bir çalışma zamanı hatasına neden olur.



Kullanmasına izin verilmeyen bir konuma erişmeye çalışan bir program düşün. Metin sorunsuzdur, ancak çalışma sırasındaki gerçek durum bir hataya neden olur. Örneğin, program özel izinler gerektiren bir sistem dosyasını okumaya çalışabilir. İşletim sistemi erişimi reddeder ve program bir hata alır. Kodun kendisi doğru olduğu için bu bir çalışma zamanı hatasıdır.



Farkı görmek için bir yazma hatası ile bir çalışma zamanı hatasını karşılaştır. Yazma hatası, yanlış yazılmış bir anahtar sözcük veya eksik bir parantez gibi dilin kurallarını ihlal eder. Bilgisayar bunu çalıştırmadan önce yakalar. Çalışma zamanı hatası ise metin doğru olsa bile eksik bir dosya gibi gerçek durumu ihlal eder. Farklı aşamalara aittirler ve farklı düzeltmeler gerektirirler.



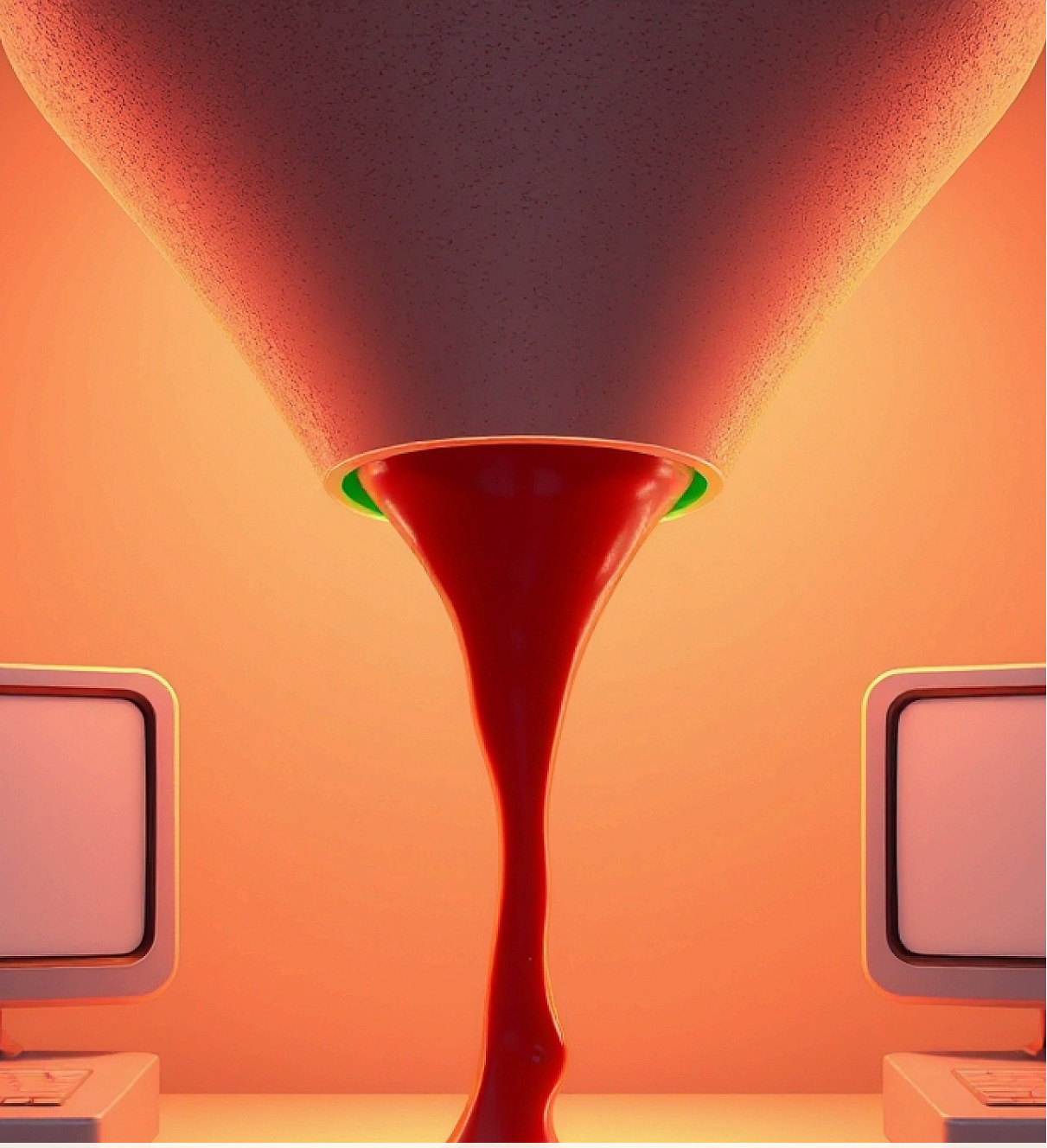
Gerçek dünya, programın beklediğinden farklı olabilir. Dosyalar eksik olabilir, konumlar yasaklı olabilir ve veriler beklenmeyen bir biçimde olabilir. Kodu ne kadar dikkatli yazarsan yaz, bu geçersiz durumlar ortaya çıkacaktır. Bu gerçeği kabul etmek önemlidir, çünkü bu, sorunların asla olmayacağını ummak yerine çalışma sırasındaki sorunlar için plan yapman gerektiği anlamına gelir.



Bu tür durumlar, program yazılırken önceden düşünülmelidir. Programcı, çalışma sırasında neyin yanlış gidebileceğini planlar. Bu, bir dosyanın var olmaması veya bir ağın çökmesi gibi olası arızaları düşünmek anlamına gelir. Bu sorunları önceden tahmin ederek, çökmek yerine onları zarifçe ele alan kod yazabilirsin. Önceden planlamak temel bir beceridir.



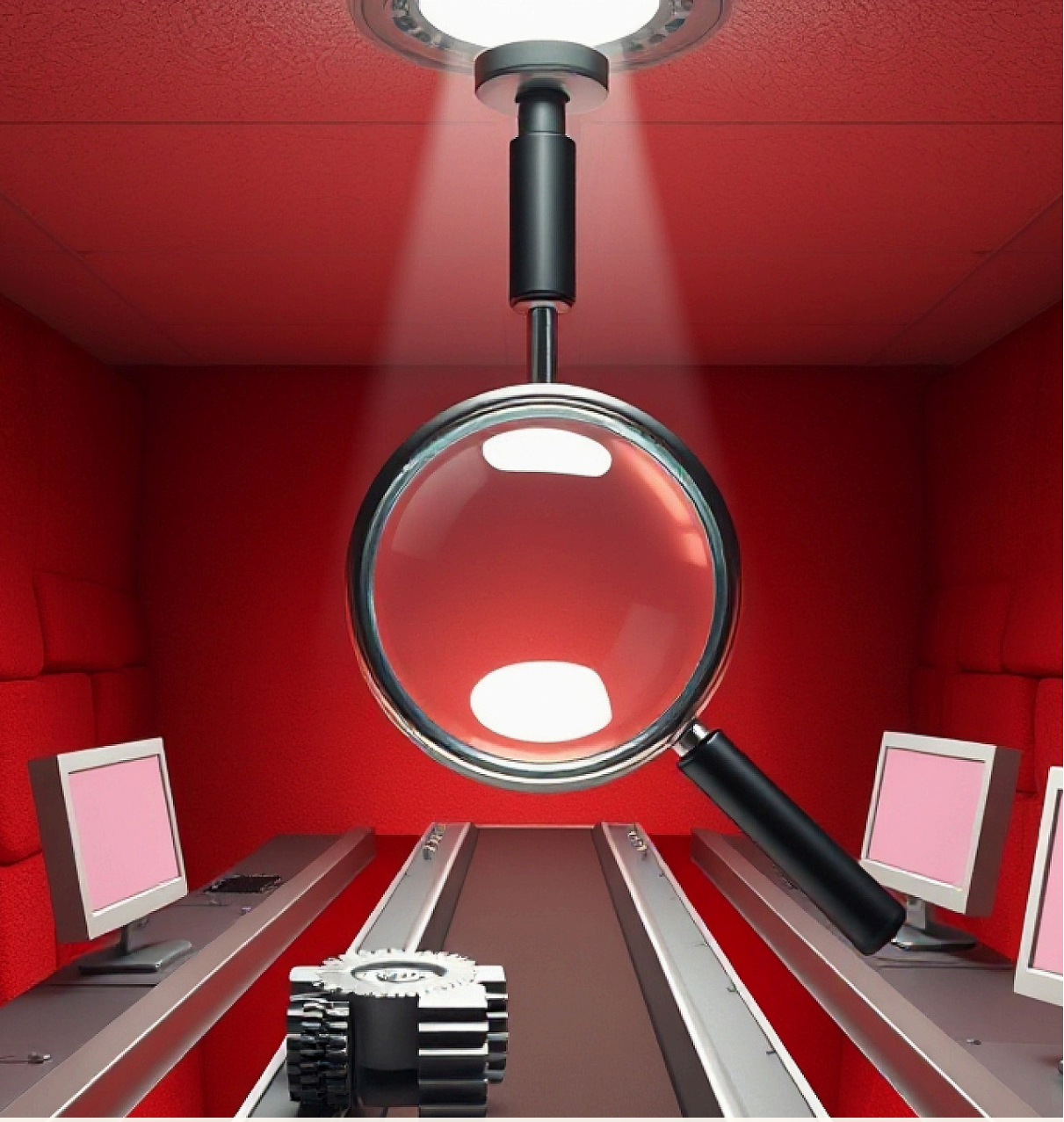
Riskli bir adımdan önce bir kontrol koy, örneğin önce bir dosyanın var olup olmadığını test et. Kontrol, geçersiz durumun bir çökmeye dönüşmesini önler. Örneğin, bir dosyayı açmadan önce program işletim sistemine dosyanın orada olup olmadığını sorabilir. Yoksa, program adımı atlayabilir veya bir mesaj gösterebilir. Bu basit kontrol bir çalışma zamanı hatasını önler.



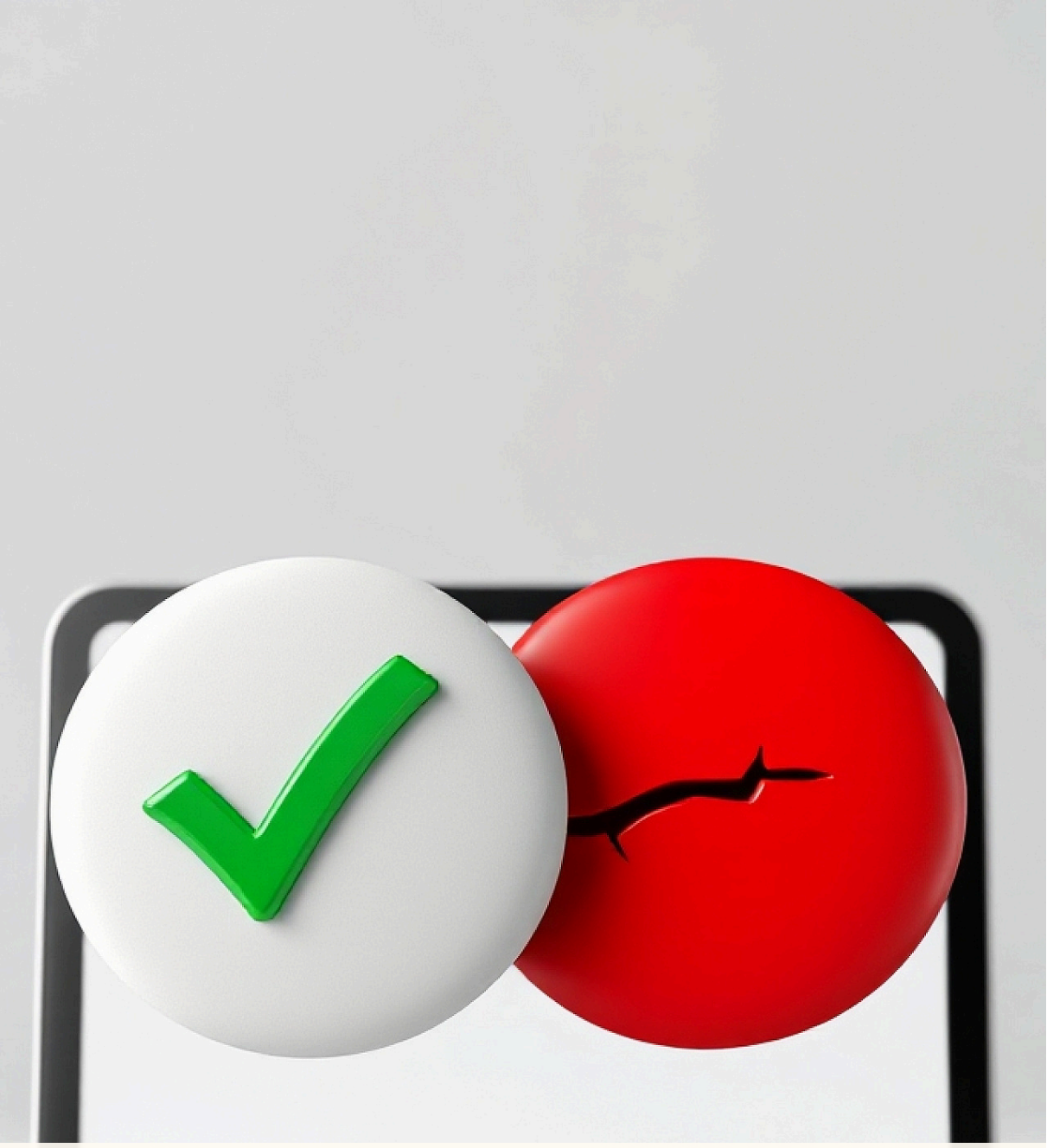
Hata yönetimi, bir sorun tespit edildiğinde ne yapılacağına karar vermek anlamına gelir. Program sadece durmaz; planlanmış bir şekilde yanıt verir. Örneğin, bir dosya açılamazsa, program sorunu günlüğe kaydedebilir ve varsayılan ayarlarla devam edebilir. Hata yönetimi, olası bir çökmeyi kontrollü bir yanıta dönüştürerek programı çalışır durumda tutar veya temiz bir şekilde çıkmasını sağlar.



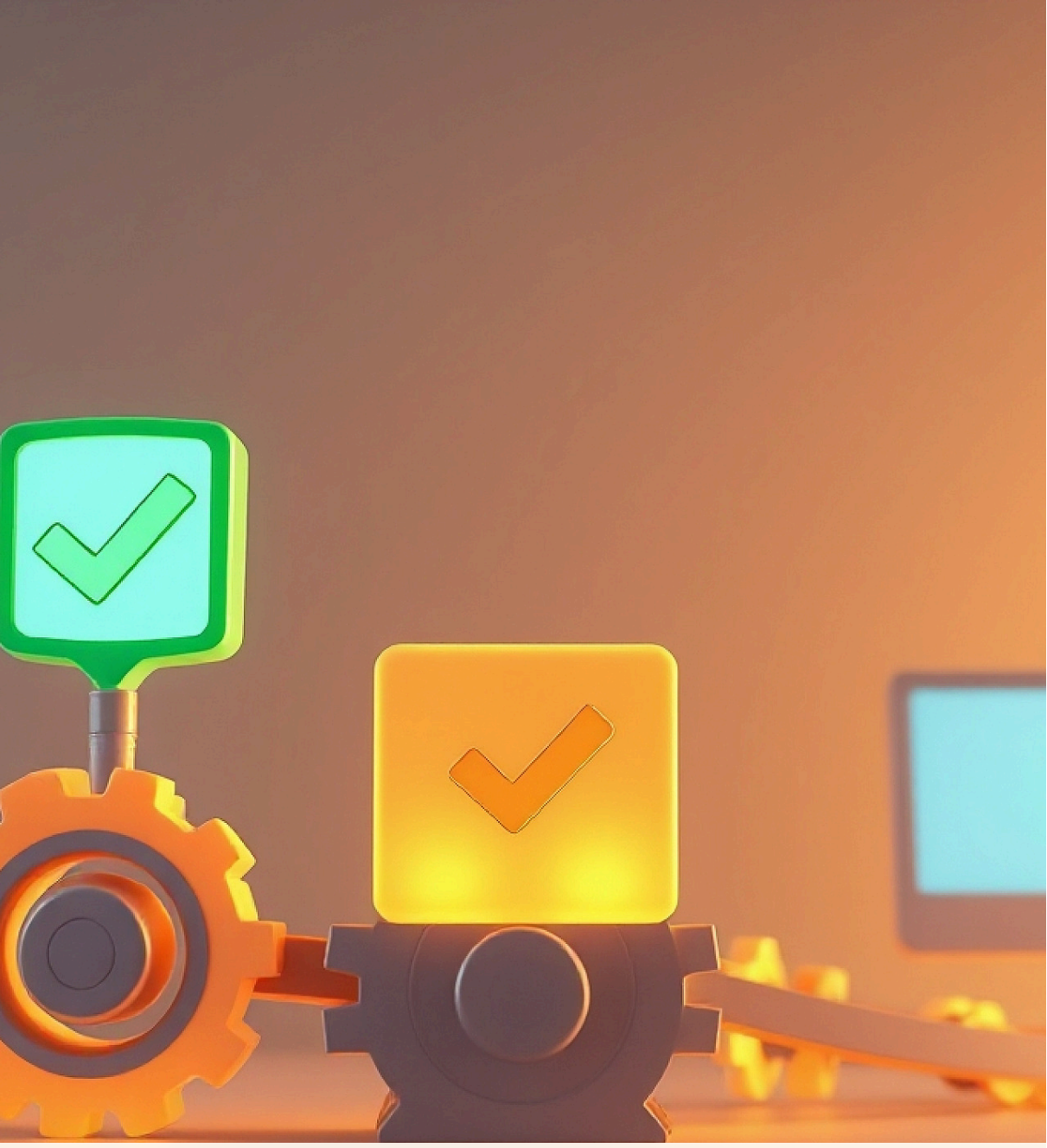
Bir sorun tespit edildiğinde, iki iyi yanıt kullanıcıya neyin yanlış gittiğini söylemek veya alternatif bir yol izlemektir. Her ikisi de programı sessizce başarısız olmak yerine kullanışlı tutar. Örneğin, bir dosya eksikse, program dosyanın bulunamadığını söyleyen bir mesaj gösterebilir veya varsayılan içerikle yeni bir dosya oluşturabilir. Her iki durumda da kullanıcı bilgilendirilir veya program devam eder.



Yeni başlayanların yaygın bir hatası, her şeyin her zaman çalışacağını varsaymak ve kontrolleri atlamaktır. Bu, gerçek dünyadaki küçük sorunları çalışma zamanı hatalarına dönüştürür. Örneğin, bir dosyayı kontrol etmeden açmaya çalışan bir program, dosya eksikse çökebilir. Kontroller ekleyerek bu tür hataları önleyebilir ve programı daha sağlam hale getirebilirsin. Her zaman neyin yanlış gidebileceğini düşün.



Bir diğer yaygın hata, her sorunu çalışma zamanı hatası olarak adlandırmaktır. Yazma hataları ile çalışma zamanı hatalarının farklı aşamalara ait olduğunu hatırla. Yazma hatası çalıştırmadan önce yakalanır, çalışma zamanı hatası ise yürütme sırasında ortaya çıkar. Bunları karıştırmak yanlış nedeni aramaya yol açabilir. Her zaman hatanın yazma sırasında mı yoksa çalışma sırasında mı oluştuğunu belirle.



Çalışma zamanı hatası, kod doğru yazılmış olsa bile geçersiz bir durum ortaya çıktığında yürütme sırasında meydana gelir. Önceden düşünmeyi, kontroller eklemeyi ve hataları ele almayı unutma; böylece program sorunu açıklayabilir veya başka bir yol izleyebilir. Beklenmeyeni planlayarak programlarını daha güvenilir ve kullanıcı dostu hale getirirsin. Çalışma zamanı hatalarını yönetmenin anahtarı budur.