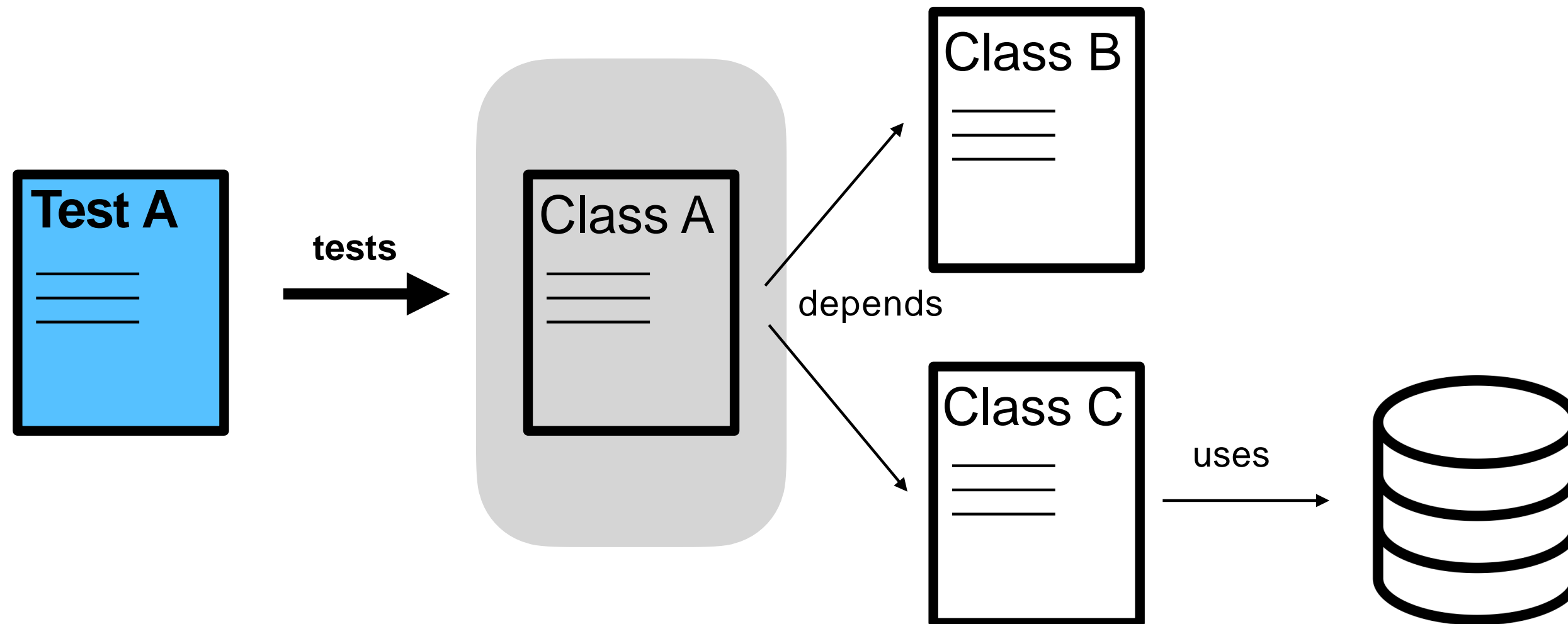


# Test Scope

# Birim Testi

Birim, bir sistemin sınıf veya bireysel yöntem gibi ayrı bir bileşenidir.

Birimlerin ayrı ayrı test edilmesine **birim testi** denir.



# Birim Testi

Birim, bir sistemin sınıf veya bireysel yöntem gibi ayrı bir bileşenidir.

Birimlerin ayrı ayrı test edilmesine **birim testi** denir.

- **Hızlı**
- **Kontrolü kolay**
- **Yazması kolay**
- **Gerçeklikten yoksun**
- **Tüm hatalar yakalanamıyor**  
(ör. diğer bileşenler veya hizmetlerle etkileşimler)

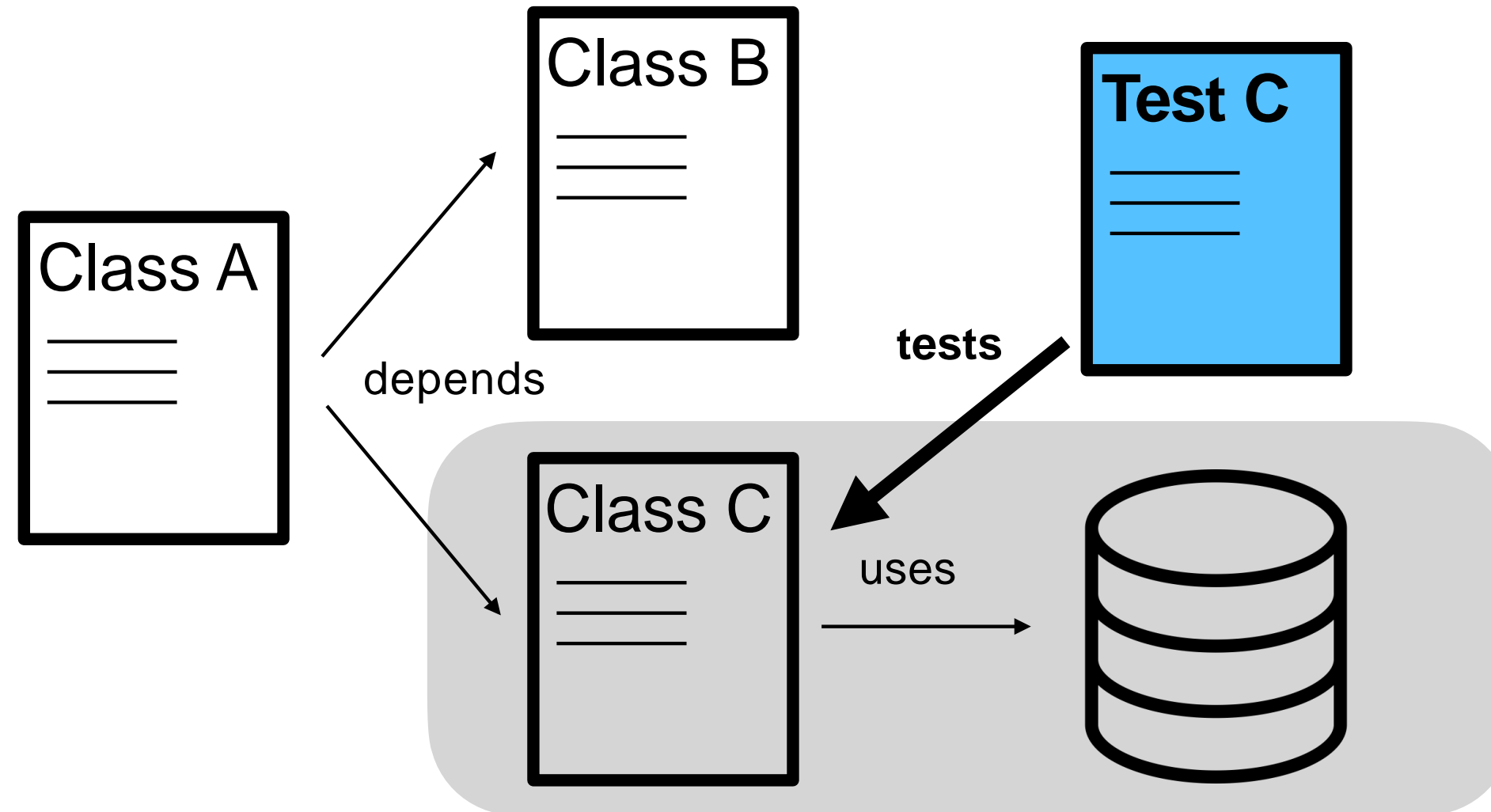
**Birim testleri çok kullanışlı bir test türüdür ancak çoğu zaman tek başına yetersizdir.**

# Entegrasyon Testleri

Tek başına test yapmak yeterli değildir. Bazen kod, sistemin sınırlarının “ötesine” gider ve diğer (çoğunlukla harici) bileşenleri (örneğin, bir veritabanı) kullanır.

**Entegrasyon testleri, bizim kodumuz ile harici tarafların kodları arasındaki entegrasyonu test eder.**

**Örnek:** Bir veritabanına SQL sorguları aracılığıyla erişen yöntemlerin test edilmesi. Yöntemlerimiz veri tabanından doğru verileri alıyor mu?



# Entegrasyon Testleri

Tek başına test yapmak yeterli değildir. Bazen kod, sistemin sınırlarının “ötesine” gider ve diğer (çoğunlukla harici) bileşenleri (örneğin, bir veritabanı) kullanır.

**Entegrasyon testleri, bizim kodumuz ile harici tarafların kodları arasındaki entegrasyonu test eder.**

**Örnek:** Bir veritabanına SQL sorguları aracılığıyla erişen yöntemlerin test edilmesi. Yöntemlerimiz veri tabanından doğru verileri alıyor mu?

- **Entegrasyon hatalarını yakalayabilir.**
- **Önem vermediğimiz bileşenler de dahil olmak üzere tüm sistemi kapsayan bir sistem testi yazmaktan daha az karmaşıktır.**

**Yazması zor, örneğin:**

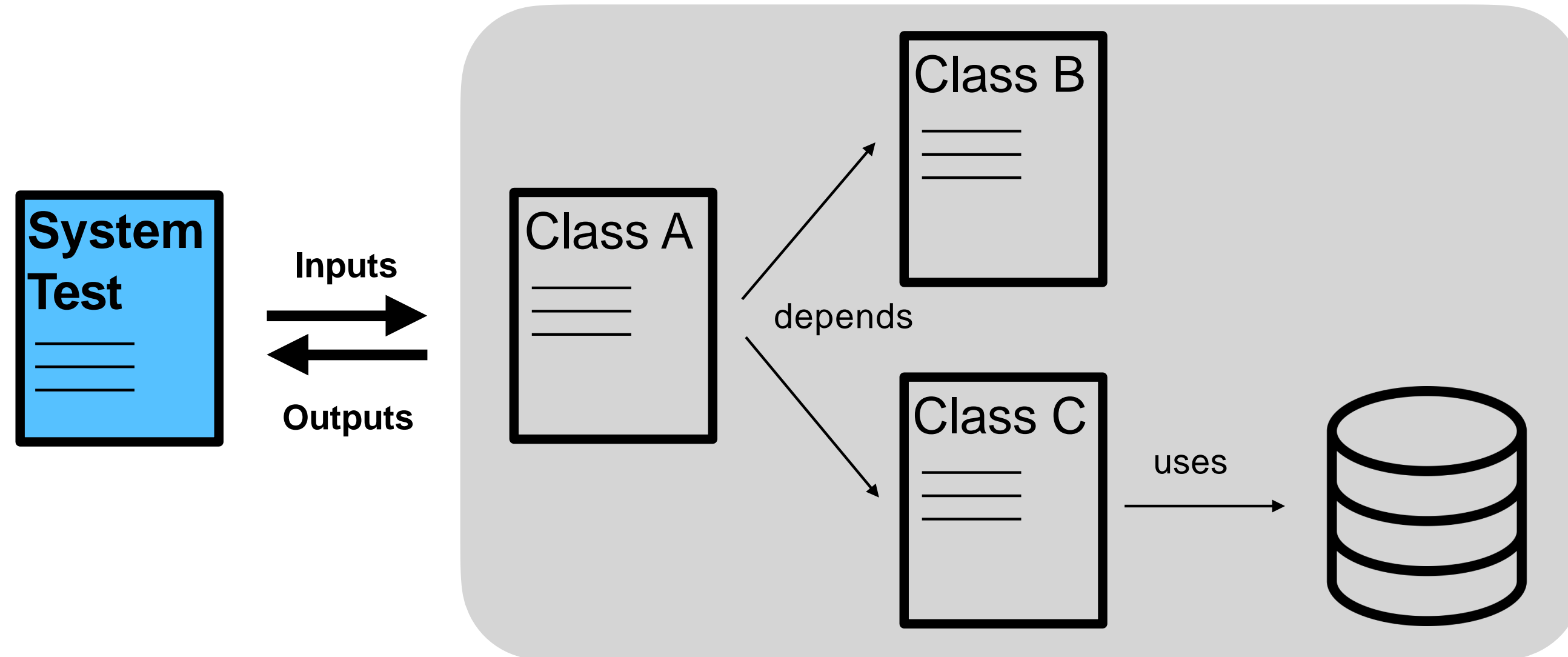
- Veritabanının yalıtılmış bir örneğini kullanmanız gerekir.
- Testin beklediği duruma getirilmeli
- Durum daha sonra sıfırlanmalı

# Sistem Testleri

Yazılımın daha gerçekçi bir görünümünü elde etmek için, tüm veritabanı, ön uç ve diğer bileşenleriyle birlikte daha gerçekçi testler yapmalıyız.

Sistemin içeriden nasıl çalıştığıyla ilgilenmiyoruz.

**Belirli girdiler verildiğinde, belirli çıktıların sistem tarafından sağlanmasına önem veriyoruz.**



# Sistem Testleri

Yazılımın daha gerçekçi bir görünümünü elde etmek için, tüm veritabanı, ön uç ve diğer bileşenleriyle birlikte daha gerçekçi testler yapmalıyız.

Sistemin içeriden nasıl çalıştığıyla ilgilenmiyoruz.

**Belirli girdiler verildiğinde, belirli çıktıların sistem tarafından sağlanmasına önem veriyoruz.**

- **Gerçekçi**

(Testler son kullanıcıya benzer performans gösterdiğinde sistemin tüm son kullanıcılar için doğru çalışacağından o kadar emin olabiliriz)

- **Yavaş!**

- **Yazması zor**

(hesaba katılması gereken çok sayıda harici hizmet)

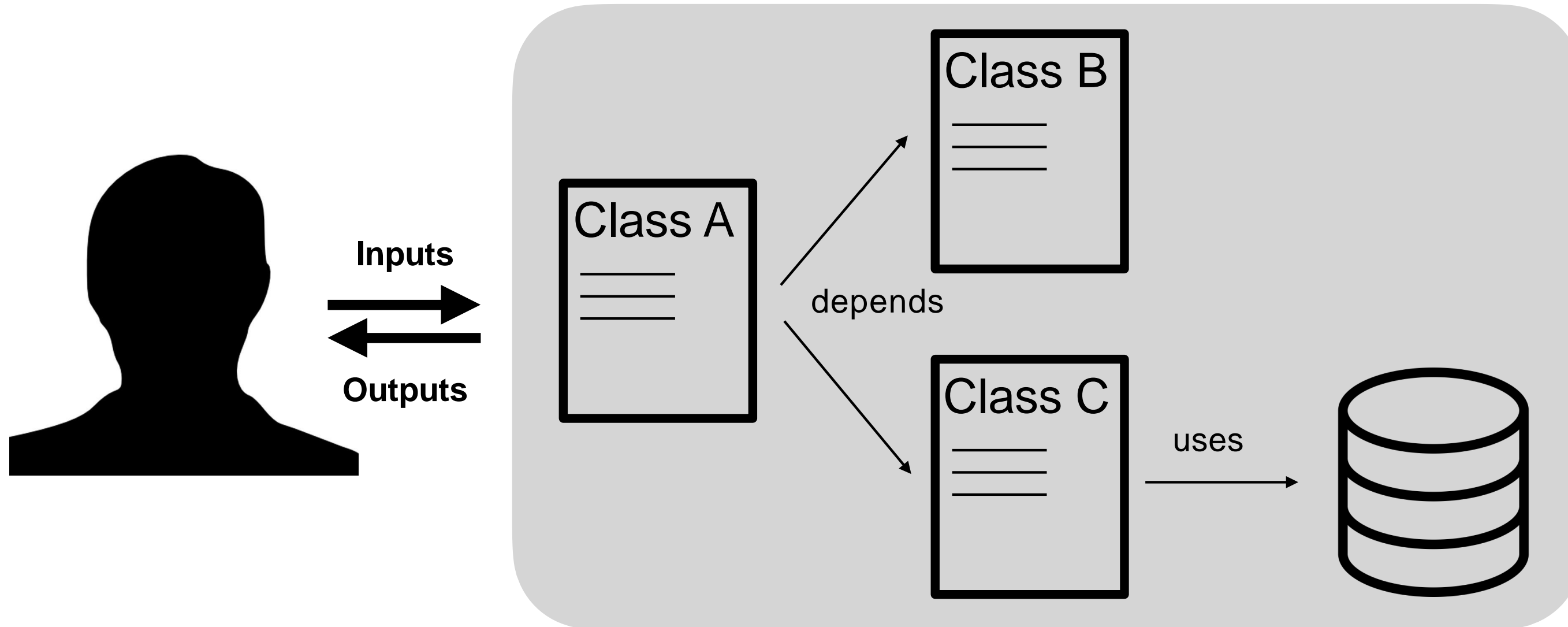
- **Flakiness eğilimli**

# Manuel Testler

Her şey otomatik bir şekilde kolayca test edilemez, özellikle de nitel yargıların olduğu durumlarda (örneğin, bir arama motorunun sonuçlarının kalitesi).

Ayrıca, hangi otomatik testlerin yazılacağını bilmek için gerçek sistem davranışını keşfetmemiz gerekebilir.

**Manuel testler, bir insan tarafından manuel olarak gerçekleştirilen sistem testleridir.**





# Manuel Testler

Her şey otomatik bir şekilde kolayca test edilemez, özellikle de nitel yargıların olduğu durumlarda (örneğin, bir arama motorunun sonuçlarının kalitesi).

Ayrıca, hangi otomatik testlerin yazılacağını bilmek için gerçek sistem davranışını keşfetmemiz gerekebilir.

**Manuel testler, bir insan tarafından manuel olarak gerçekleştirilen sistem testleridir.**

- **Gerçek**

(Test cihazı, sistemi fiilen kullanan bir son kullanıcı olarak hareket ediyor)

- **Zaman alıcı**

- **Tekrarlaması zor**

- **Sıkıcı**

# Test Üçgeni

