

# Metodlar

## Uygulamalı Kod

1. Genişlik ve yükseklik parametrelerini alan ve bu değerlerle dikdörtgenin alanını hesaplayan `calculateRectangleArea` adında bir Java programı oluşturun.
2. Belirli bir deseni yazdıran `printPattern` adında bir Java programı tasarlayın. Örneğin, `n = 4` olarak verildiğinde, program aşağıdaki deseni yazdırmalıdır:

```
1
12
123
1234
```

3. Verilen bir sayının asal olup olmadığını kontrol eden bir Java uygulaması geliştirin. Bir tam sayı alan ve eğer asal bir sayıysa `true`, değilse `false` döndüren `isPrime` adında bir metod oluşturun.
4. Verilen bir dizesini tersine çeviren bir Java programı yazın. Girdi olarak bir dize alan ve tersine çevrilmiş dizesini döndüren `reverseString` adında bir metod kullanın.
5. Sıcaklık birimlerini Celsius'ten Fahrenheit'e ve Fahrenheit'ten Celsius'e dönüştüren bir Java programı uygulayın. Dönüşüm hesaplamaları için `celsiusToFahrenheit` ve `fahrenheitToCelsius` adında metodlar oluşturun.

Önceki eksiklik için özür dilerim. İşte yalnızca Java'da metodların temellerine odaklanan gözden geçirilmiş çalışma soruları:

## Çalışma Soruları

1. Java'da bir metodun amacı nedir? Metodların kodu düzenleme ve yapılandırmada oynadığı temel rolü açıklayın.

**Cevap:** Java'da bir metod, belirli bir görevi gerçekleştiren ve birden fazla kez kullanılabilen bir kod bloğudur. Metodlar kodu parçalara ayırarak daha yönetilebilir hâle getirir, modülerlik, okunabilirlik ve tekrar kullanılabilirliği artırır. Metodlar işlevselliği kapsüller, programlamada yapılandırılmış ve düzenli bir yaklaşımı teşvik eder.

2. Java'da bir metodun bileşenlerini açıklayın. Metod adı, dönüş tipi, parametreler ve metod gövdesinin bir metodu tanımlamadaki önemini açıklayın.

**Cevap:**

- **Metod Adı:** Metodu tanımlar ve çağırmak için kullanılır. İsimlendirme kurallarına uymalı ve açıklayıcı olmalıdır.
- **Dönüş Tipi:** Metodun döndüreceği değerin tipini belirtir. Metod hiçbir şey döndürmeyecekse `void` olabilir.
- **Parametreler:** Metoda iletilen değerleri temsil eder. Metodun girdisini tanımlar, opsiyonel olabilir.
- **Metod Gövdesi:** Metodun çalıştığında ne yapacağını belirten kod bloğunu içerir.

3. Metod bildirimi ile metod çağrısının farkını tartışın. Bir metodun çağrılması bir metodu tanımlamaktan nasıl farklıdır?

**Cevap:**

- **Metod Bildirimi:** Bir metodun adını, dönüş tipini, parametrelerini ve metod gövdesini tanımlar.
- **Metod Çağrısı (veya Metod Çağırısı):** Metodu adıyla kullanma veya yürütme eylemidir; gerekiyorsa parametrelerle birlikte. Metod çağrısı, metodu tanımlamaktan farklı olarak, metod gövdesindeki kodu çalıştırır.

4. Java'da metod parametrelerinin önemi nedir? Parametreler nasıl tanımlanır ve metodlar içinde nasıl kullanılır? Önemlerini anlamak için örnekler verin.

**Cevap:** Metod parametreleri, bir metoda iletilen değerleri depolayan değişkenler olarak işlev görür. Metodların farklı değerler almasını ve dinamik olarak kullanmasını sağlar. Parametreler metod bildiriminde tanımlanır ve metodun davranışını etkileyebilir. Örneğin:

```
void mesajYaz(String mesaj) {  
    System.out.println(mesaj);  
}
```

Burada, `String mesaj`, `mesajYaz` metodunun farklı mesajları almasını ve yazdırmasını sağlayan bir parametredir.

5. Java metodlarında **return** ifadesinin rolünü açıklayın. Bir değeri bir metoddan nasıl döndürmek için kullanılır ve bir **return** ifadesiyle karşılaşıldığında ne olur?

**Cevap:** Java'da **return** ifadesi, bir metodun çalışmasını sonlandırmak ve (metodun void olmayan bir dönüş tipi varsa) belirtilen bir değeri çağırıcıya iletmek için kullanılır. **return** ifadesiyle karşılaşıldığında, metodun yürütülmesi durur ve belirtilen değer (varsa) çağırıcıya aktarılır. Ayrıca, kontrolü çağırının olduğu yere geri verir.

6. Metod bildirimlerinde **void** anahtar kelimesinin önemi nedir? Ne zaman ve neden kullanılır? Kullanımını göstermek için örnekler verin.

**Cevap:** **void** anahtar kelimesi, bir metodun herhangi bir değer döndürmediğini belirtir. Bir metodun sonuç üretmesine gerek olmadığında kullanılır. Örneğin:

```
void selam() {  
    System.out.println("Merhaba!");  
}
```

Burada **void**, `selam`

metodunun herhangi bir değer döndürmediğini belirtir.

7. Java programlamada metodların kullanımının faydaları nelerdir? Metodların kodun tekrar kullanılabilirliğine, okunabilirliğine ve bakımına nasıl katkıda bulunduğunu tartışın.

**Cevap:**

- **Kodun Tekrar Kullanılabilirliği:** Metodlar, aynı işlevselliği kodu yeniden yazmadan birden çok kez kullanmamızı sağlar.
- **Okunabilirlik:** Kodu küçük, isimlendirilmiş metodlara bölmek, genel kodun daha okunabilir ve anlaşılır olmasını sağlar.
- **Bakım Kolaylığı:** Metodlar, kodu modüler hale getirerek bakım, hata ayıklama ve güncelleme süreçlerini kolaylaştırır.

8. Java'da metod isimlendirme için gelenekler ve en iyi uygulamaları açıklayın. Daha iyi kod anlaşılması için anlamlı ve açıklayıcı metod isimleri sağlamak için hangi kurallara uyulmalıdır?

**Cevap:** Metod isimleri açıklayıcı ve anlamlı olmalı, camelCase adımlama kurallarına uygun olmalı ve metodun eylemini veya amacını doğru bir şekilde yansıtmalıdır. Eylemleri tanımlamak için fiiller kullanılmalı, belirsiz isimlerden kaçınılmalı ve açıklık ve tutarlılık için çaba gösterilmelidir.