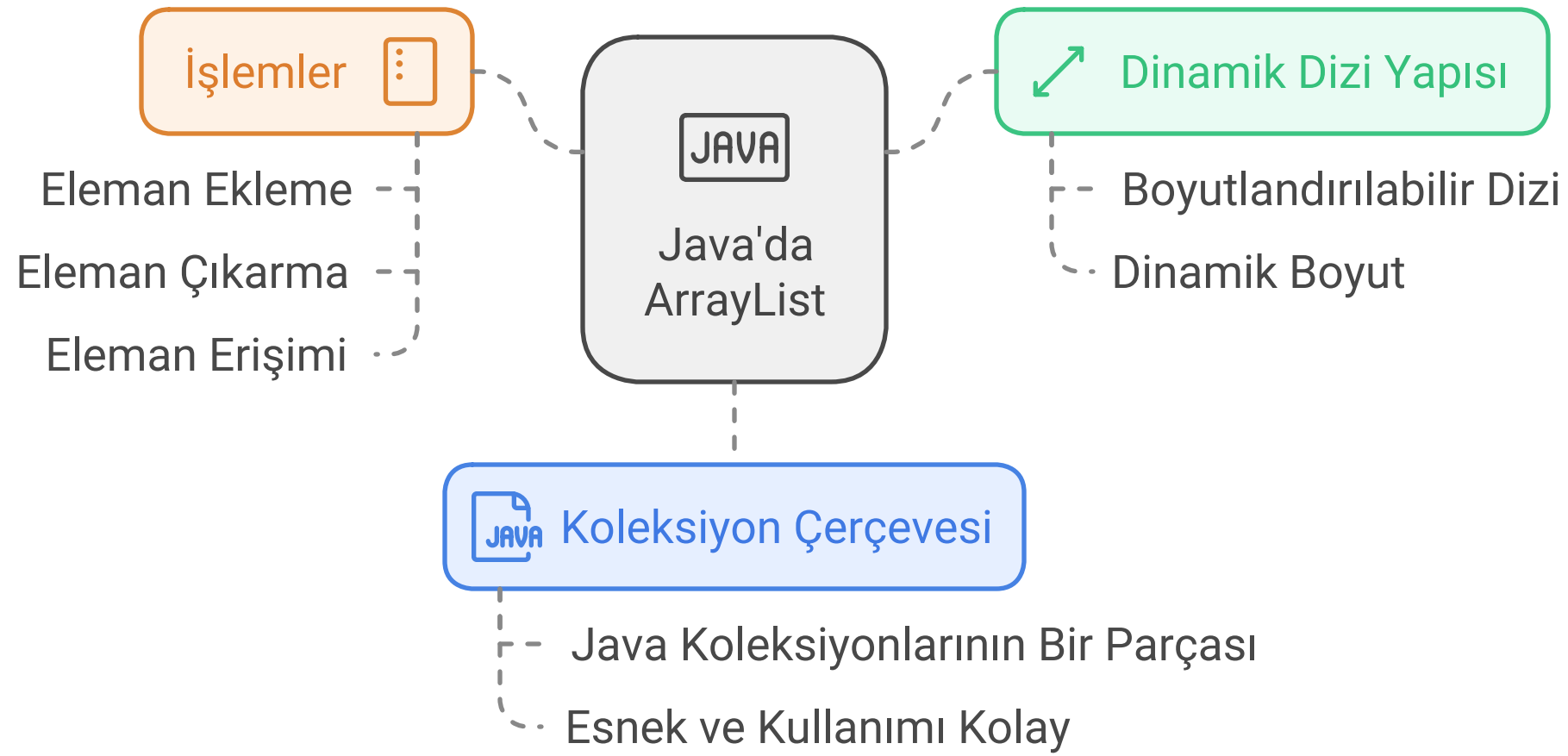


Java'da ArrayList Konusu

Bu belgede, Java programlama dilinde kullanılan ArrayList veri yapısını adım adım inceleyeceğiz. ArrayList, dinamik bir dizi yapısıdır ve eleman ekleme, silme gibi işlemleri kolaylaştırır. Java'nın Collection Framework'ünün bir parçası olan ArrayList, esnekliği ve kullanımı kolay olması nedeniyle yaygın olarak tercih edilir. Bu belgede, ArrayList'in temel özellikleri, nasıl oluşturulacağı, eleman ekleme ve çıkarma işlemleri gibi konular detaylı bir şekilde ele alınacaktır.



1. ArrayList Nedir?

ArrayList, Java'da dinamik bir dizi yapısıdır. Normal diziler sabit boyutludur, ancak ArrayList boyutunu dinamik olarak değiştirebilir. Bu, eleman ekleme ve silme işlemlerinin daha esnek bir şekilde yapılmasını sağlar.

ArrayList vs. Dizi

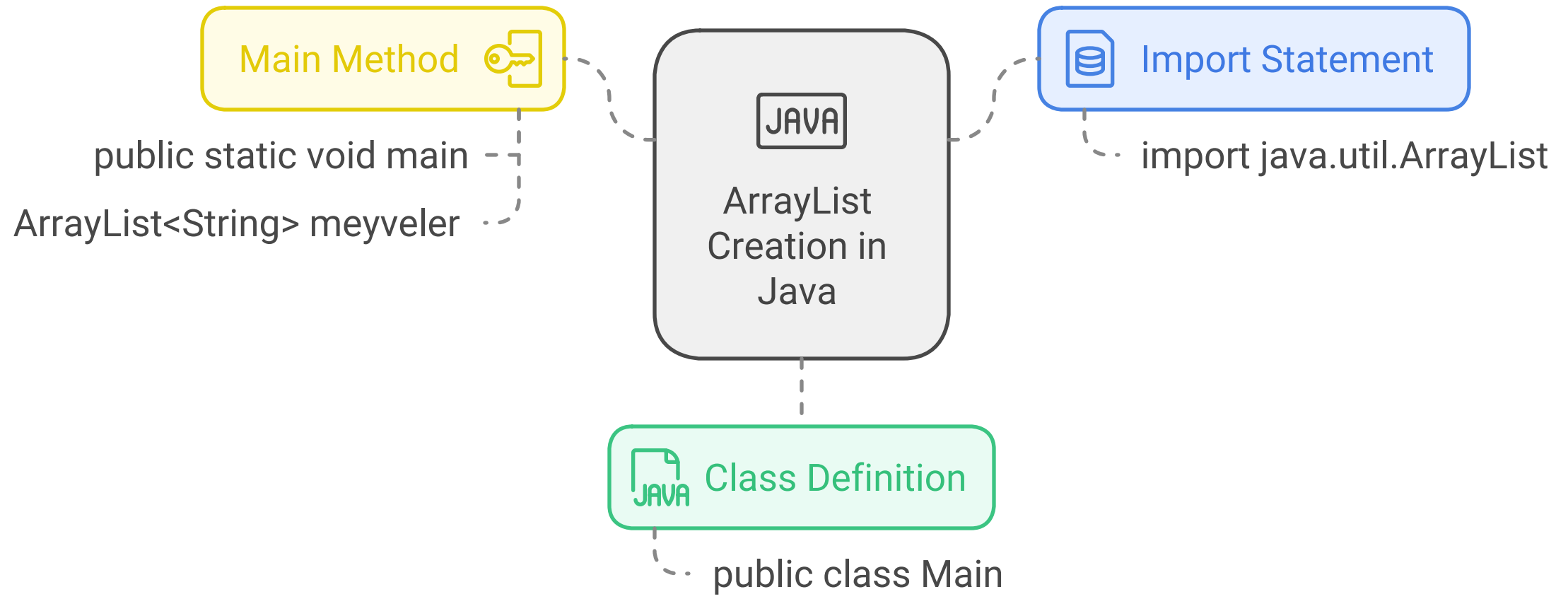


2. ArrayList Oluşturma

ArrayList oluşturmak için **java.util** paketinden **ArrayList** sınıfını kullanmalıyız. Aşağıda basit bir ArrayList oluşturma örneği verilmiştir:

```
import java.util.ArrayList;

public class Main {
    public static void main(String[] args) {
        ArrayList<String> meyveler = new ArrayList<>();
    }
}
```

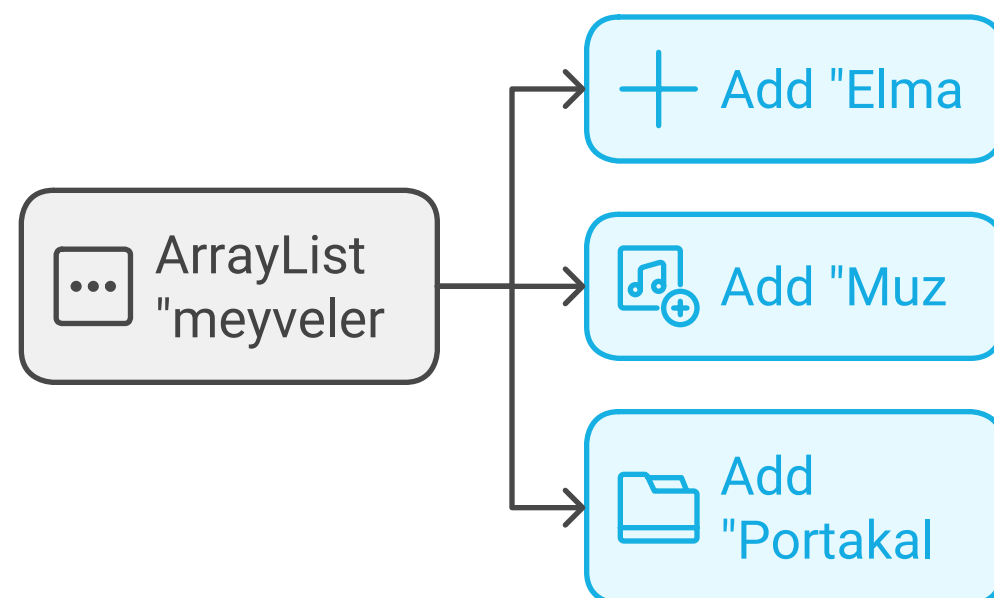


Yukarıdaki kodda, **String** türünde bir ArrayList oluşturduk. Bu listeye sadece **String** türündeki elemanları ekleyebiliriz.

3. Eleman Ekleme

ArrayList'e eleman eklemek için **add()** metodunu kullanırız. Aşağıda bir örnek verilmiştir:

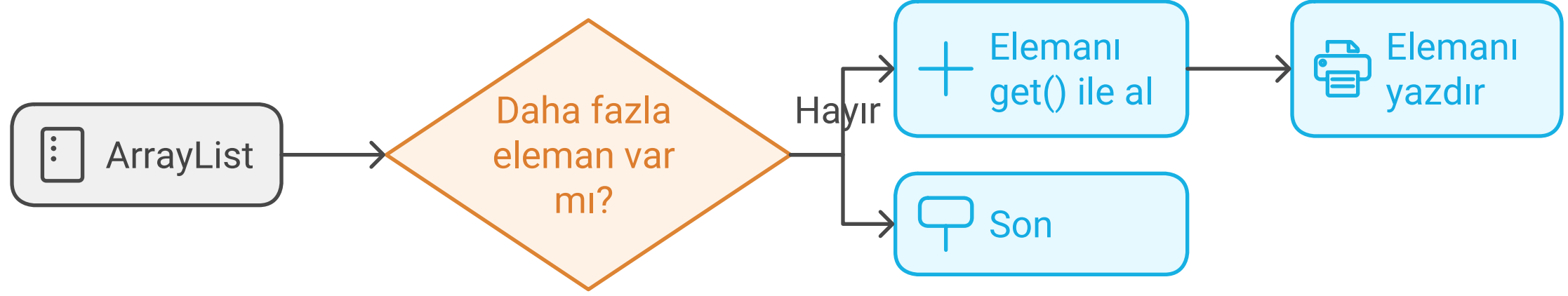
```
meyveler.add("Elma");
meyveler.add("Muz");
meyveler.add("Portakal");
```



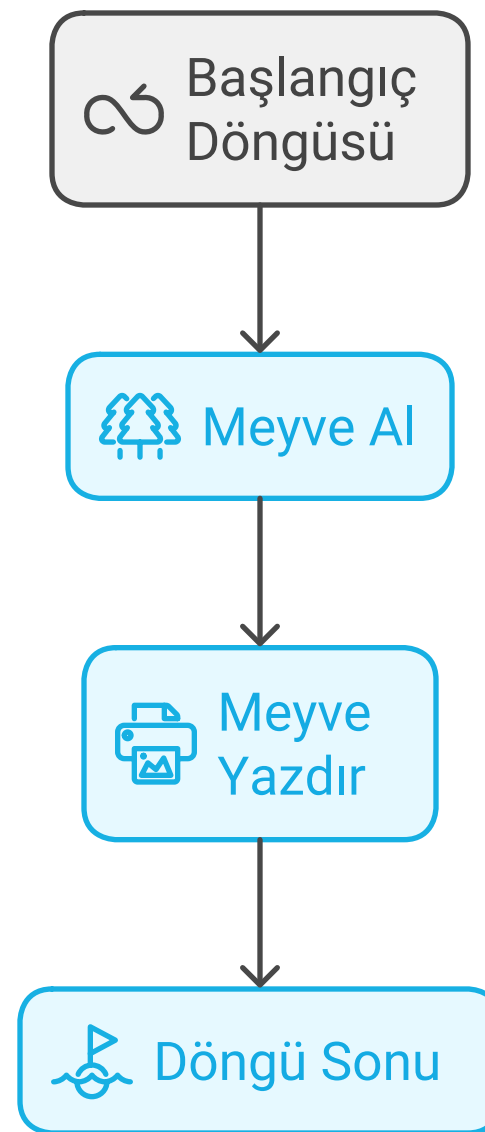
Bu kod, **meyveler** ArrayList'ine üç farklı meyve ekler.

4. Elemanları Görüntüleme

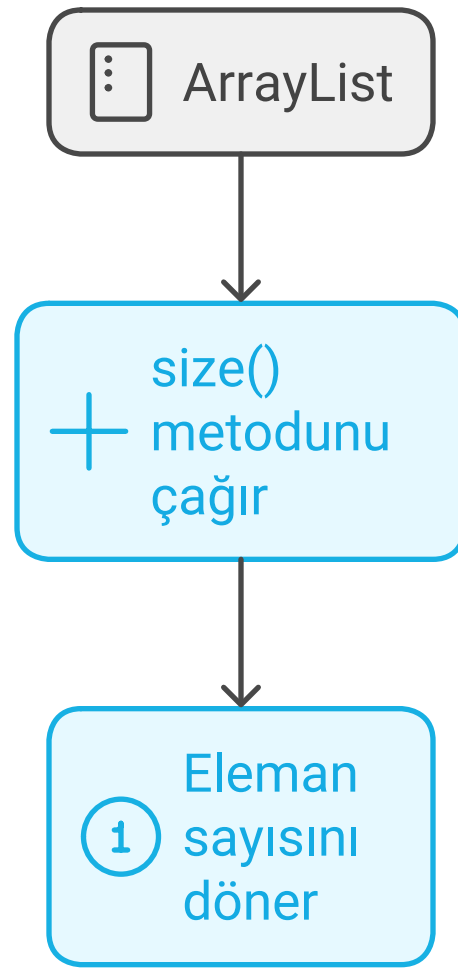
ArrayList'teki elemanları görüntülemek için **get()** metodunu kullanabiliriz. Aşağıda tüm elemanları yazdıran bir örnek bulunmaktadır:



```
for (int i = 0; i < meyveler.size(); i++) {  
    System.out.println(meyveler.get(i));  
}
```



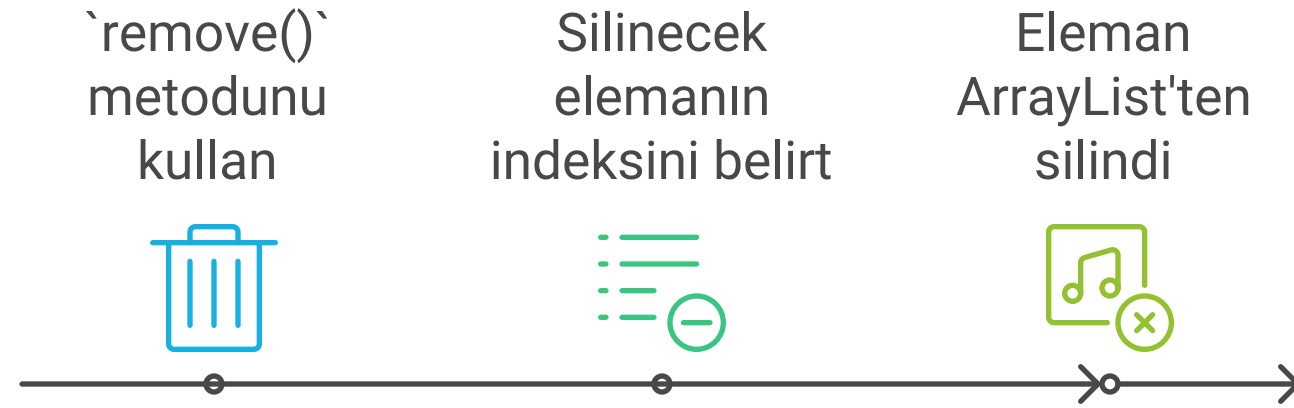
size() metodu, ArrayList'in mevcut eleman sayısını döner.



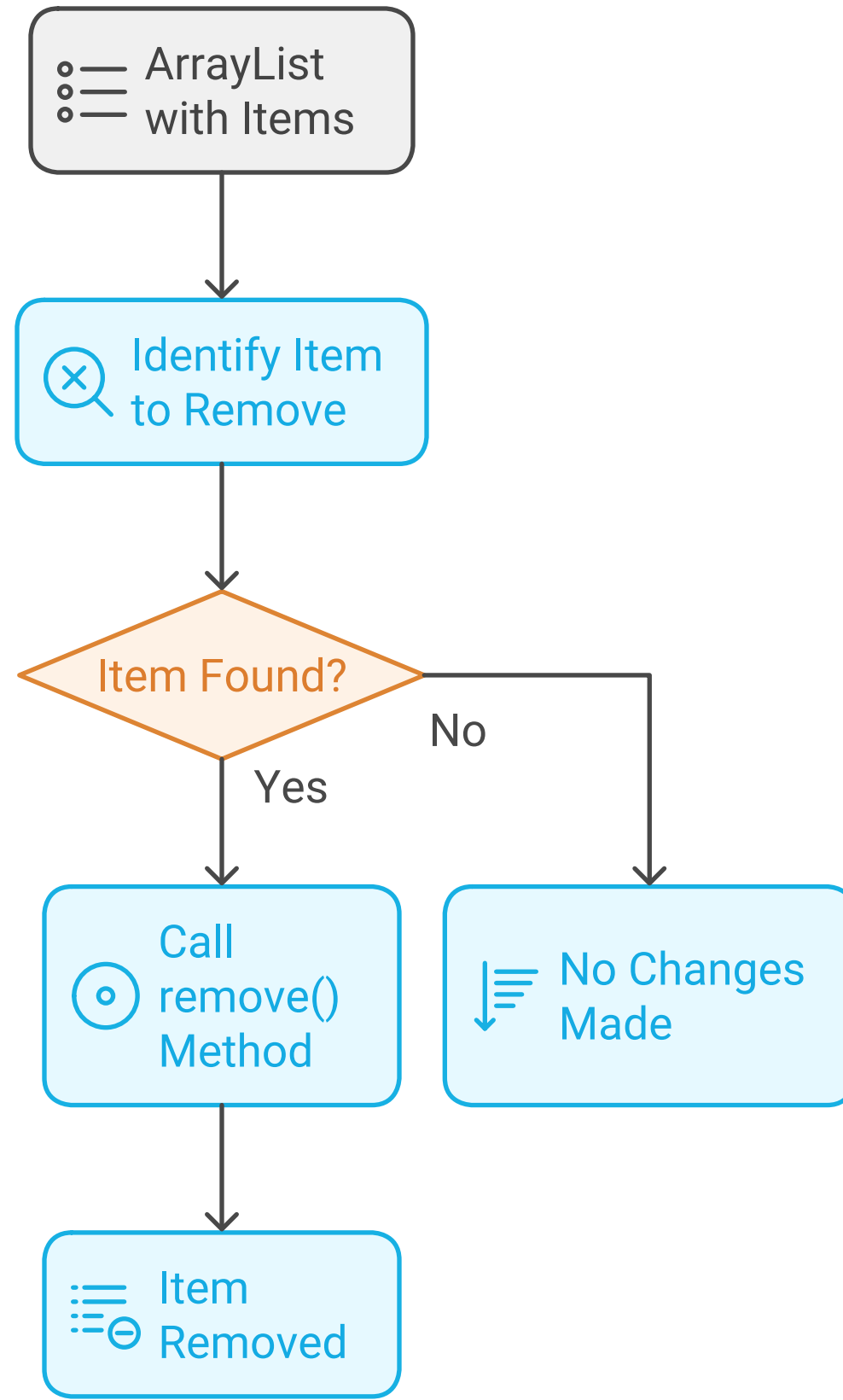
5. Eleman Silme

`ArrayList`'ten eleman silmek için **`remove()`** metodunu kullanırız. Aşağıda bir örnek verilmiştir:

Bir `ArrayList`'ten bir eleman sil



```
meyveler.remove("Muz");
```



Bu kod, **meyveler** listesinden "Muz" elemanını siler.

6. Eleman Sayısını Öğrenme

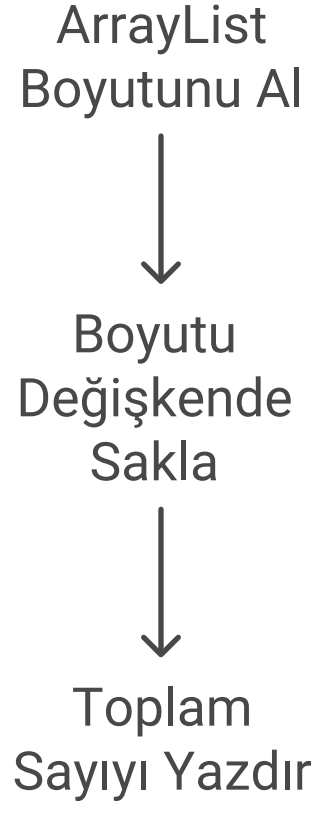
ArrayList'teki eleman sayısını öğrenmek için **size()** metodunu kullanabiliriz:

Java'da bir ArrayList'in boyutunu nasıl alırsınız?



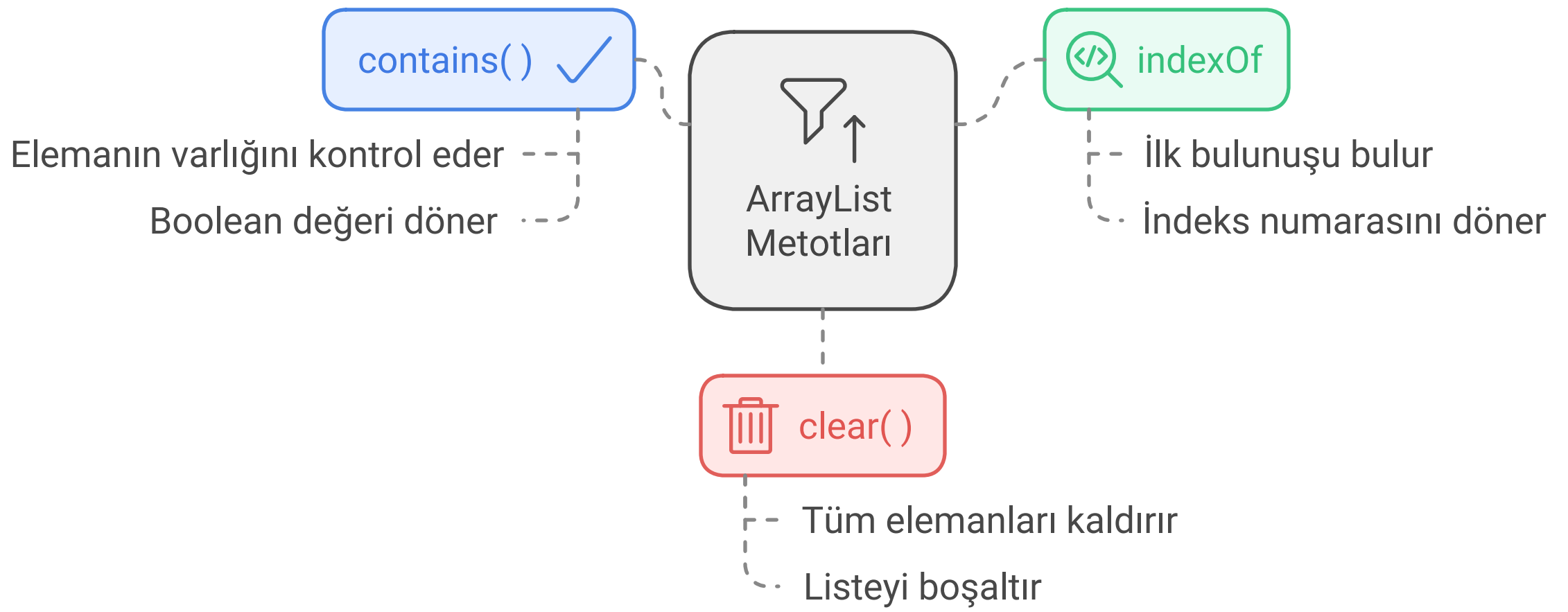
```

int toplamMeyve = meyveler.size();
System.out.println("Toplam meyve sayısı: " + toplamMeyve);
  
```



7. ArrayList'in Diğer Özellikleri

ArrayList, birçok faydalı metod sunar. Bunlardan bazıları:



- **contains(Object o)**: Belirtilen elemanın listede olup olmadığını kontrol eder.
- **indexOf(Object o)**: Belirtilen elemanın ilk bulunduğu indeks numarasını döner.
- **clear()**: Tüm elemanları siler.

8. Sonuç

Java'da ArrayList, dinamik veri yapısı olarak oldukça kullanışlıdır. Eleman ekleme, silme ve görüntüleme işlemleri oldukça kolaydır. Bu belgede, ArrayList'in temel özelliklerini ve kullanımını adım adım inceledik. Java programlamada ArrayList kullanarak veri yönetimini daha etkili bir şekilde gerçekleştirebilirsiniz.