

ArrayList ve LinkedList Java'nın Collection Framework'ünde bulunan iki farklı liste implementasyonudur. Her iki sınıf da bir liste (veri yapıları) sunar, ancak içsel yapıları ve performans özellikleri açısından farklılıklar gösterir. İşte bu iki sınıf arasındaki temel farklar:

1. İçsel Veri Yapısı:

ArrayList: Dinamik olarak büyüyen bir dizidir. Elemanları birbirine bağlı olarak değil, ardışık bellek bloklarında saklar. Bu nedenle, eleman eklemek veya çıkarmak için diziyi kaydırmak gerekebilir.

LinkedList: Çift yönlü bir bağlı listedir. Her eleman, bir önceki ve bir sonraki elemanı gösteren referanslara sahiptir. Bu nedenle, eleman ekleme veya çıkarma işlemleri daha hızlıdır çünkü sadece ilgili referansları güncellemek yeterlidir.

2. Performans:

ArrayList: Elemanlara doğrudan erişim, dizinin indeksine dayalı olduğu için hızlıdır. Ancak, eleman eklemek veya çıkarmak, dizinin boyutunu yeniden ayarlamayı gerektirebilir, bu da performansı etkileyebilir.

LinkedList: Eleman ekleme veya çıkarma işlemleri daha hızlıdır çünkü sadece bağlantıları güncellemek yeterlidir. Ancak, doğrudan indeksleme yoktur, bu nedenle belirli bir indekse gitmek daha uzun sürebilir.

3. Bellek Kullanımı:

ArrayList: Bellek kullanımı genellikle daha düşüktür çünkü sadece elemanları içeren bir diziyi saklar.

LinkedList: Her eleman için bir önceki ve bir sonraki elemanı gösteren referanslar olduğu için daha fazla bellek kullanır.

4. Kullanım Senaryoları:

ArrayList: Elemanlara hızlı bir şekilde erişmek ve sıralı verilere ihtiyaç duyulan durumlar için uygundur.

LinkedList: Eleman ekleme veya çıkarma işlemleri sık yapılan durumlar için uygundur. Özellikle büyük veri setleri üzerinde sıklıkla yapılan bu tür işlemlerde daha etkilidir.

Genel olarak, kullanım senaryolarına ve performans gereksinimlerine bağlı olarak ArrayList veya LinkedList tercih edilebilir.