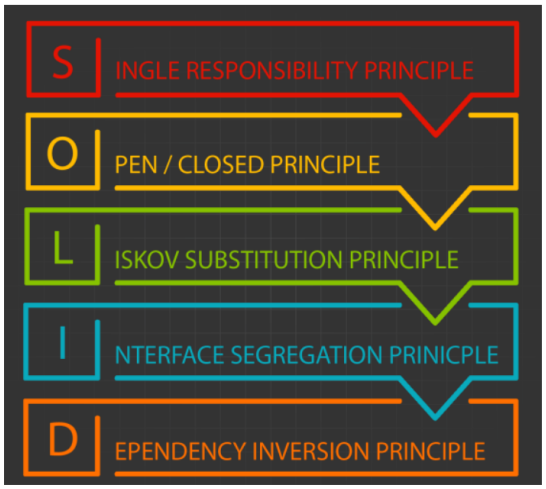
Selinay Altun

Java Bootcamp Vize Soruları

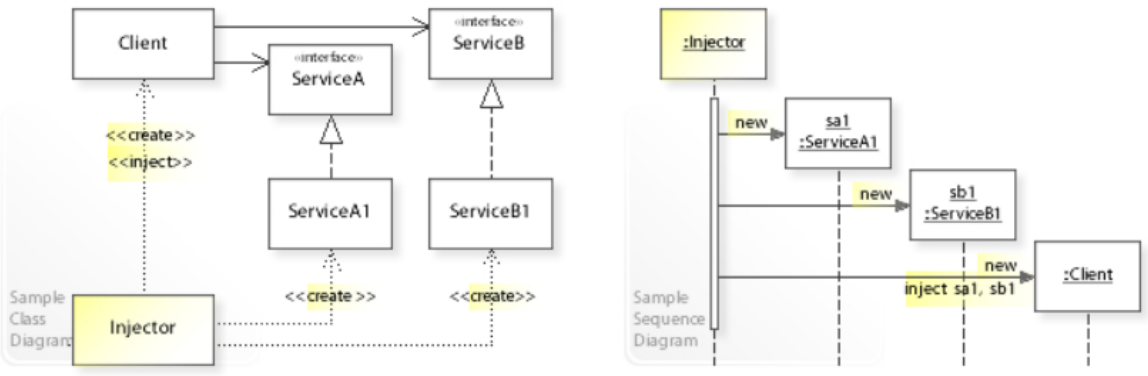
1) Dependency Injection

SOLID**,** nesne tabanlı programla ile yazılım geliştirilirken kullanılan beş önemli tasarım ilkesidir.Bu tasarım ilkesinin son prensibi Dependency Inversion ile tasarlanan üst sınıflar ile alt sınıflarının bağımlı olmamasıdır,bağımlılığın en aza indirilmesidir.



Şekil 1:SOLID

Bunu gerçekleştirmek için çeşitli teknikler vardır.Bu prensiplerden biri ise Dependency Injection(DI)dir.Dependency Injection, yazılımın bağımlılıklardan kurtulmasını amaçlayan ve olabildiğince bağımsızlaştıran bir programlama tekniğidir.Bunu ise bağımlılık oluşturabilecek nesneleri dışardan vererek direkt kullanılması önlenir ve programdaki bağımlılık azaltılır. Nesne yönelimli programlamada bağımlılığın az olması bir nesnede ya da sınıfta yapılan geliştirmelerin veya değiştirmelerin programda problemler yaratmadan esnek bir şekilde yapılmasını için çok önemlidir.



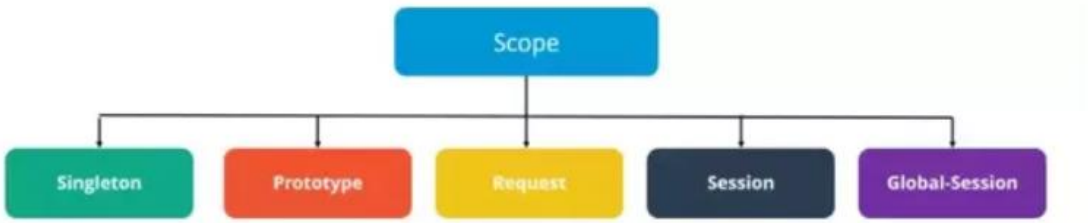
Şekil 2:Dependency Injection

Üst sınıfların alt sınıflara direkt bağlı olmadan soyutlama kullanılarak bağımlılığının azalmasıdır.Bağlılığın az ya da gevşek tutulmasına ise Loosely Coupled denir.Ayrıca Dependency Injection sayesinde Unit test yazılımı kolaylaşır ve bağımlılığı azaltılan sınıflar tek başına test edilebilir.

Constructor Injection, Method Injection,Interface Injection olmak üzere üç farklı kullanım alanı vardır. Constructor Injection:Bağımlılık, constructor ile sağlanır. Method Injection:bağımlılık, setter methodu ile sağlanır. Interface Injection:Bağımlılık, setter methoduna sahip interface implement edilerek sağlanır.

2) Bean Scope

Bean, Spring Framework de temeli oluşturan ve yeniden kullanılabilir nesnelere denir.Bean, Spring IOC container ile yönetilir.Bean Scope ise bean nesnelerinin yaşam alanlarıdır.Bean scope kullanım alanlarına göre belirlemek gerekir.Birçok Scope türü vardır.



Şekil 3:Bean Scope Türleri

Singleton : Varsayılan olarak tanımlanan Scope.Bean'den yalnızca bir tane üretilir.

Prototype : Yeni nesne, oluşturma isteği ile oluşturulur.Her oluşturmada farklı oluşturulur.

Request : HTTP isteği geldiğinde oluşturulur. Web ile uyumlu application dosyası oluşturulduğunda kullanılabilir.

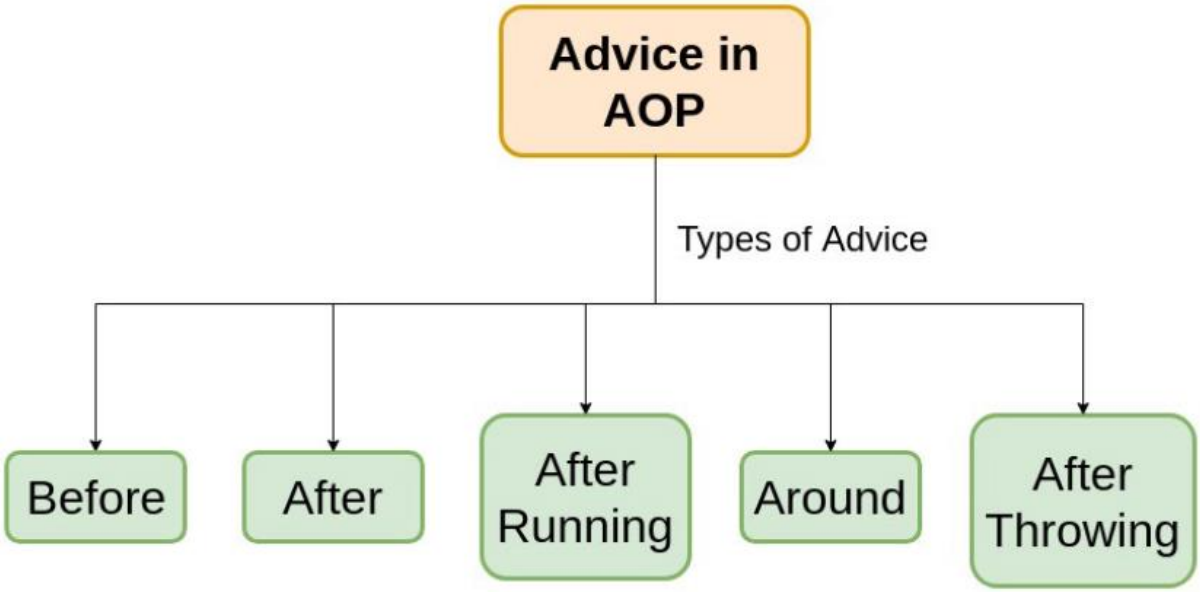
Session : Web uygulamalarında oturum geçerli olduğu sürece kullanılabilir.HTTP session'ı için bir adet bean oluşturulur.

Global-Session : HTTP'nin yaşam döngüsünde tek bir Bean'ın tanımını kapsamaktadır. Web uygulamaları için kullanılır.

3) Pointcut, Advice

Pointcut, kısaca Join Point topluluğu ya da kümesi denebilir.Join Point ise kodların execute edileceği zamanın belirtildiği ön koşullardır. Pointcut, advice yerine getirilmesi gerekip gerekmediğini belirlemek için join point ile eşleştirilen ifadelerdir. org.aspectj.lang.annotation kullanılır. Target, methodun çalışması hedeflenen beani ifade eder.Execution, methodun çalışacağı bölge belirtilir.

Advice, belirli bir join point için gerçekleştirilen eylemlerdir. Programlama açısından, uygulamada pointcut ile eşleşen belirli bir birleştirme noktasına ulaşıldığında çalıştırılan yöntemlerdir.

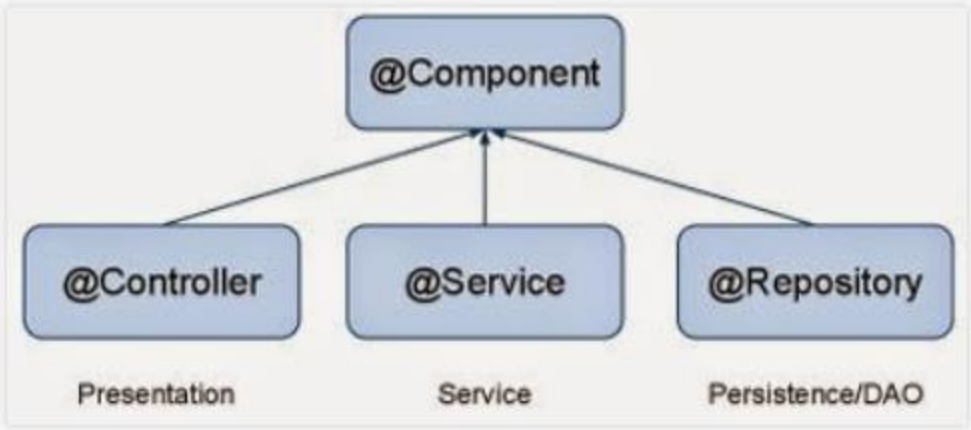


Şekil 4:Advice

@Before: Method devreye girmeden önce çalışır.
@AfterReturning: Method başarılı olduktan sonra çalışır.
@AfterThrowing: Methodun exception dönmesi durumunda çalışır.
@After: Returning ve Throwing her iki durumdada çalışır.
@Around: Method devreye girmeden önce ve method bittikten sonra çalışır.

4) @Repository and @Controller

Spring'deki bazı anotasyonlardan birileridir.Peki anotasyon nedir?Anotasyonlar runtime anında framework tarafından yorumlanan ifadelerdir.Tanım yapar yazılım geliştirme sürecini kolaylaştırır ve hızlandırır.



Şekil 5:Repository and Controller

Sınıfın üstüne @Repository ve @Controller anotasyonları yazıldığı zaman bu sınıf Spring Bean olarak kaydedilir ve Spring tarafından yönetilir.Bu anotasyonlar sayesinde her sınıf için xml ayarlarının yapılmasına gerek kalmaz.

@Repository: Spring 2.0 ile kullanılmaya başlanmıştır.Dao katmanında kullanılır.Dao katmanında ise genellikle database işlemleri yapılmaktadır.

@Controller: Spring 2.5 ile kullanılmaya başlanmıştır.MVC sınıflarında kullanılmaktadır.

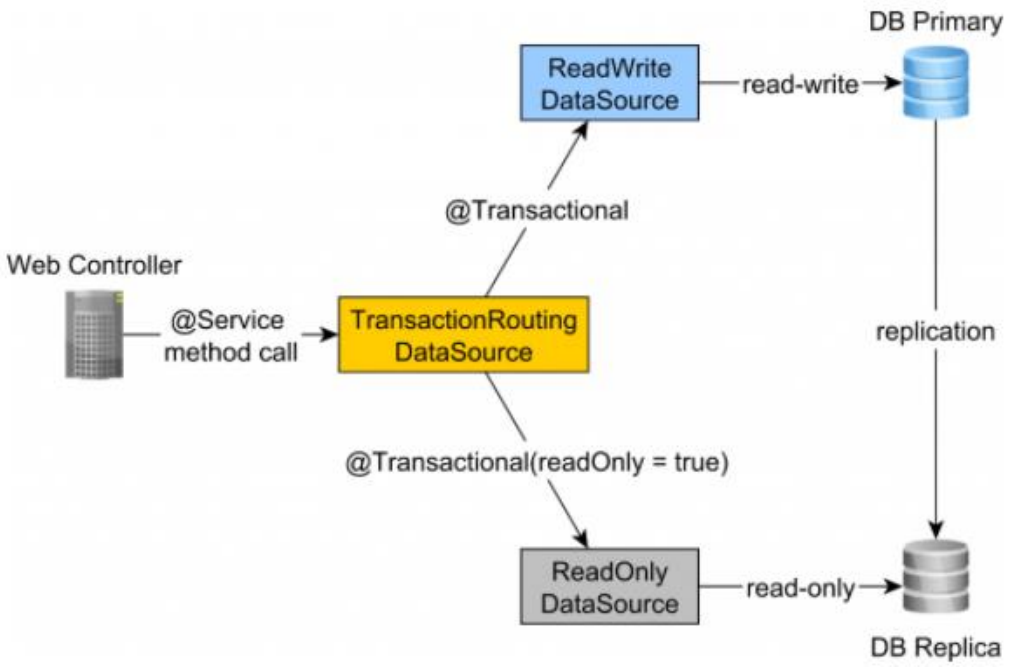
5) @Autowired

Spring 2.5 ile kullanılmaya başlanmıştır.Temelde Dependency Injection mantığına uygundur. @Autowired anotasyonu ile bean framework kontrolü ile başka bir sınıfa inject edilebilir.@Autowired ile nesne bağımlılıkları kolaylıkla yönetilmiş olunur.

@Autowired anotasyonu sayesinde daha az kod yazılır ve üst-alt sınıf bağımlılığını azaltılır.Ancak programcanın kontrol edemeyeceği durumlar da oluşabilir.

6) Transaction

Transaction işlemi veri tabanında gerçekleştirilen bir veya birden fazla sorguların (SQL) bir bütün olarak işleme alınmasına denir.Bütün olarak alınan işlemlerin hepsi başarlı olursa aynı anda commit edilir yani onaylanır.Ancak herhangi biri başarısız olursa bütün bu işlemi rollback yapılır yani geri alınır. Transaction işlemi sayesinde veri bütünlüğü sağlanır. Transaction işleminde yapılan herhangi bir değişiklik bütün sql sorguları için geçerli olur.



Şekil 6:Transaction

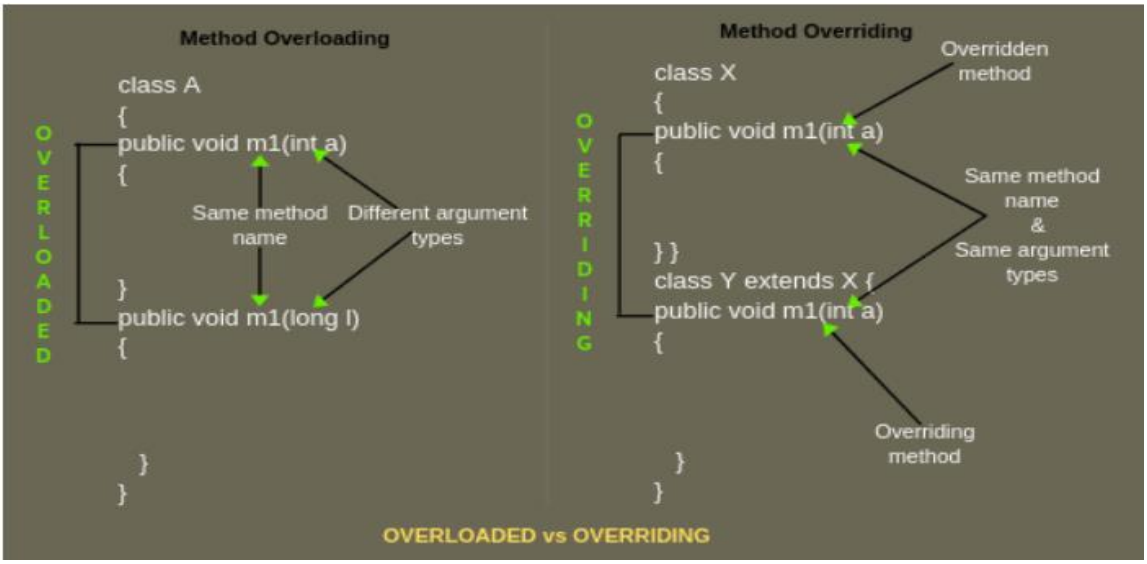
Transaction işlemi Spring'de ise transaction yönetim API ile sağlanır. Birbirinden farklı veri erişim yöntemlerini aynı anda kullanmayı sağlar. Spring'de diğer işlemlerde de kullanılan anotasyon kullanılarak bu işlem kolay ve doğru bir şekilde gerçekleştirilir.

Spring, transaction işlemi için Programatik ve Deklaratif olmak üzere iki tane yöntem sunar. Programatik yöntem ile Transaction işleminin başlatılıp sonlandırılması gibi kodları barındırır.Deklaratif yöntem ise Spring'in sahip olduğu kurallar ile Spring Container tarafından kontrol edilir.Deklaratif yöntemde @Transactional anahtar kelimesi kullanılarak veritabanındaki işlemlerin yönetimi Spring kontrolünde ilerler.İşlem,class ya da method düzeyinde gerçekleştirilebilir.Eğer class düzeyinde ise @Transactional anotasyonu tüm puclic methodları kapsar.Methodlar başarılı olduğunda işlem commit edilir yani onaylanır.

7) Overriding and Overloading

Overload, kelime anlamı aşırı yükleme demektir.Bir methodun aynı isimle farklı parametrelere sahip olarak birden fazla çalışacak şekilde yazılmasına overloading denir.Overloading sayesinde aynı işlemi farklı parametrelerle yapması gerektiği durumlarda kolaylıkla metotlar oluşturmayı sağlar.Aynı isimle oluşturulan bu metotları

birbirinden ayrıyan şey aldıkları farklı parametrelerdir.Bu şekilde derleyici hata vermeden doğru bir şekilde çalışır.Eğer aynı sayıda ve aynı sırada aynı parametreler yazılırsa program hata verir.Burada en önemli fark parametredir, örneğin aynı isimli ve aynı parametreli fakat farklı dönüş değerli metotlar da hata verir.



Şekil 7:Overloading and Overridin

Overriding,kelime anlamı ezme ya da geçersiz kılmaktır. Üst sınıftan alınan mirasla gelen methodun geçersiz kılınmasıdır.Alt sınıf miras aldığı bu methodu istediği gibi yani gövdesini değiştirerek kullanılır.Bu methodun ismi ve parametreleri aynı kalır ancak gövde değiştirilir.Miras aldığı methodu değiştirerek kendi içinde aynı methodu farklı yorumlamasıdır.

Farkları ise:

- Overloadingde parametreler farklıdır ancak overridingde parametreler ve sıraları aynıdır.
- Overloadingde gövde aynıdır ancak overridingde gövde farklıdır.
- Overloading işleminde kullanılacak olan metot derleme zamanında belirlenirken Overriding işleminde ise kullanılacak metot çalışma zamanında belirlenir ve bu yüzden hataları düzenlemek daha zordur.

8) ArrayList and Vector

ArrayList, java.util paketinde bulunan ve Collections Framework parçası olan yapılardır.ArrayList,dinamik bir dizi yapısında olan yani eleman eklendiğinde büyüyen ancak çıkarıldığında küçülen kendi içinde özel methodları bulunan kolay ve hızlı kod yazmayı sağlayan yapılardır.Dizidir ancak boyut sınırlaması yoktur.

Ekleme yapıldığı zaman sırayı korur,silinme işlemi yapıldığı zaman kaydırma yapılması gerekir.Ekleme işlemi O(n), arama işlemi O(1) ve silme işlemi ise O(n) zaman karmaşasına sahiptir.İlkel türlerden (int,char vb.) ArrayList oluşturulamaz.Boyutu önceden belirlenemeyen ve dinamik yapıda depolama alanı olması gereken zamanlarda ArrayList kullanılır.

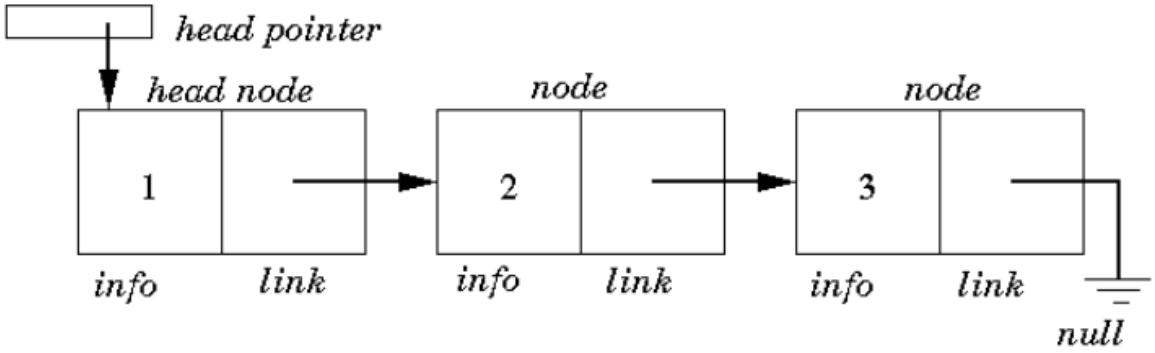
Vector, Vector sınıfını kullanır. Vector de eklenen eleman sırasını korur. Senkronize edilir. Bu nedenle, çok sayıda iş parçacığı bir vektör üzerinde aynı anda çalışamaz.

Farkları ise:

- ArrayList, senkronize olmayan veri yapısıdır yani aynı anda birden fazla iş parçacığının aynı anda çalışmasına izin verirken, Vector ise senkronize olan veri yapısıdır yani aynı anda birden fazla iş parçacığının aynı anda çalışmasına izin vermez.
- ArrayList'te yapılan işlemler Vector'de yapılan işlemlere göre çok daha verimlidir ve pergormansı güçlüdür.
- ArrayList'te yapılan işlemler Vector'de yapılan işlemlere göre çok daha hızlıdır.

9) Linkedlist

LinkedList, Java Collections Framework altında List yapısının bir parçası olan veri yapılarıdır. Türkçe karşılığı Bağlı Listelerdir.LinkedList,t utulması gereken verinin yanında bir de kendinden sonra gelecek olan node adresini de tutan ve bu verileri RAM üzerinde tutar.Bir sonraki node'un yerinin bilinmesi bellekte sıralama konusunda esneklik sağlar.Tek yönlü, çift yönlü, circular ve hybrid olmak üzere çeşitleri de vardır.Bazı dezavantajları vardır.Ramdom Access e izin vermez ve bir sonraki node'un adresini de bellekte tutması fazladan bellek kaybına neden olmaktadır.

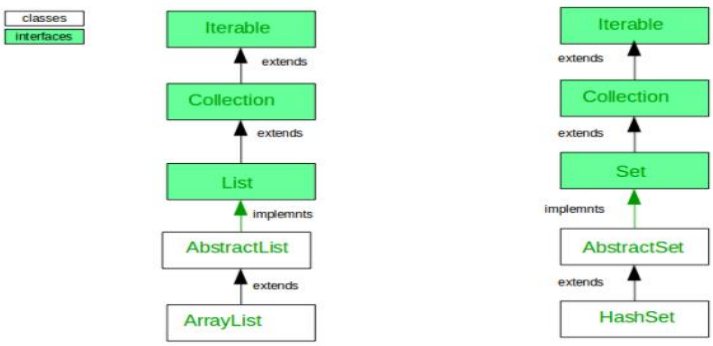


Şekil 8:LinkedList

Linked List birçok avantajlara sahiptir.Bunlardan bazıları ise hafızayı dinamik olarak kullanır.ArrayList'in aksine ekleme yapılacağı zaman kaydırma işleminin yapılamasına gerek yoktur.Sürekli büyümesi gerekn ya da ara elemanlara ekleme yapılması gerektiği zamanlarda LinkedList kullanılması daha verimli olur.Ekleme ve silme işlemleri sık sık yapılacaksa yine LinkedList kullanımı ArrayList kullanımına göre çok daha mantıklıdır.Başa ve sona ekleme işlemleri de çok daha hızlıdır.

10) ArrayList , HashSet, TreeSet

- ArrayList, List arabirimini uygularken HashSet ise Set arabirimini uygular.
- ArrayList, Array tarafından desteklenirken HashSet ise HashMap tarafından desteklenir.
- ArrayList tekrarlanan değerlere izin verirken HashSet tekrarlanan değerlere izin vermez.
- ArrayList,eklemeler yapılırken sıra korunurken HashSet ise sırasız bir şekilde ekleme yapar.



Şekil 9:ArrayList vs HashSet

- HashSet, verileri depolamak için HashMap kullanırken TreeSet verileri toplamak için TreeMap kullanır.
- Bir kıyaslama ile sıralama yapılacaksa TreeSet , HashSet'e göre çok daha avantajlıdır.
- HashSet verilerin eklenmesi,silinmesi için O(1) zaman karmaşıklığı varken TreeSet verilerin eklenmesi, silinmesi için O(log(n)) zaman karmaşıklığı vardır.
- HashSet'in performansı TreeSet'e göre çok daha verimlidir ve hızlı çalışır.
- HashSet bir kere de olsa bir null değere izin verir ancak TreeSet asla null değere izin vermez.